



TUGAS AKHIR - TE 145561

**RANCANG BANGUN *GENERIC PROGRAMMABLE*
PENGHUBUNG PENGGUNA DENGAN I/O DARI ARDUINO**

Setyo Budi Utomo
NRP 2214030052

Dosen Pembimbing
Ir. Djoko Suprajitno Rahardjo, M.T
Dr. Ir. Achmad Affandi, DEA

PROGRAM STUDI KOMPUTER KONTROL
Departemen Teknik Elektro Otomasi
Fakultas Vokasi
Institut Teknologi Sepuluh Nopember
Surabaya 2017



FINAL PROJECT - TE 145561

***RANCANG BANGUN GENERIC PROGRAMMABLE
PENGHUBUNG PENGGUNA DENGAN I/O DARI ARDUINO***

Setyo Budi Utomo
NRP 2214030052

Advisor
Ir. Djoko Suprajitno Rahardjo, M.T
Dr. Ir. Achmad Affandi, DEA

COMPUTER CONTROL STUDY PROGRAM
Electrical and Automation Engineering Department
Vocational Faculty
Institut Teknologi Sepuluh Nopember
Surabaya 2017

PERNYATAAN KEASLIAN TUGAS AKHIR

Dengan ini saya menyatakan bahwa isi sebagian maupun keseluruhan Tugas Akhir saya dengan judul "**Rancang Bangun *Generic Programmable* Penghubung Pengguna Dengan I/O Dari Arduino**)" adalah benar-benar hasil karya intelektual mandiri, diselesaikan tanpa menggunakan bahan-bahan yang tidak diijinkan dan bukan merupakan karya pihak lain yang saya akui sebagai karya sendiri.

Semua referensi yang dikutip maupun dirujuk telah ditulis secara lengkap pada daftar pustaka.

Apabila ternyata pernyataan ini tidak benar, saya bersedia menerima sanksi sesuai peraturan yang berlaku.



Rendra Kurnia .R
NRP 2214030051

Surabaya, 9 Juni 2017



Setyo Budi Utomo
NRP 2214030052

-----Halaman ini sengaja dikosongkan-----

**RANCANG BANGUN *GENERIC PROGRAMMABLE*
PENGHUBUNG PENGGUNA DENGAN I/O DARI ARDUINO**

TUGAS AKHIR

**Diajukan Guna Memenuhi Sebagian Persyaratan
Untuk Memperoleh Gelar Ahli Madya
Pada**

**Program Studi Komputer Kontrol
Departemen Teknik Elektro Otomasi
Fakultas Vokasi
Institut Teknologi Sepuluh Nopember**

Menyetujui:

Dosen Pembimbing I

Dosen Pembimbing II

Ir. Djoko Suprajitno Rahardjo, M.T.
NIP. 19550622 198701 1 001

Dr. Ir. Achmad Affandi, DEA.
NIP. 19651014 199002 1 001

**SURABAYA
JUNI, 2017**

-----Halaman ini sengaja dikosongkan-----

RANCANG BANGUN *GENERIC PROGRAMMABLE* PENGHUBUNG PENGGUNA DENGAN I/O DARI ARDUINO

Nama : Rendra Kurnia .R

Nama : Setyo Budi Utomo

Pembimbing 1 : Ir. Djoko Suprajitno Rahardjo, M.T.

Pembimbing 2 : Dr. Ir. Achmad Affandi, DEA

ABSTRAK

Memprogram perangkat Arduino dewasa ini membutuhkan kemampuan untuk menguasai Bahasa program Arduino itu sendiri, itu membuat orang awam yang ingin merakit sensor dengan input output kesulitan untuk menemukan pengendali mikro *single-board* yang bersifat open source dengan *budget* yang murah. Terpaksa mereka harus belajar Bahasa pemrograman Arduino mulai dari awal dan juga dalam menentukan *library* sensor harus mencari *library* yang cocok dengan tipe input dan output yang akan di hubungkan ke arduino dan itu sangat tidak efisien sekali.

Tugas Akhir ini merupakan penerapan dari *software* yang kami buat untuk memudahkan pengguna dalam memprogram arduino tidak perlu belajar Bahasa perograman lagi hanya menggunakan software yang kami buat dan kami beri nama “*Do It By Yourself*” yang memudahkan proses pemrograman, hanya dengan menentukan tipe Arduino dan jenis *input* dan *output* yang telah di sediakan maka *software* akan otomatis menanamkan *library* sensor kedalam Arduino sesuai dengan *input output* yang di tentukan tadi sekaligus menuliskan bahasa pemrogramannya.

Dalam *software* ini kami hanya membatasi beberapa *input* dan *output* saja karena keterbatasan kemampuan *support* komponen, dalam hal ini *software* kami aplikasikan untuk memprogram pada proses sistem keamanan dan monitoring suhu dan kelembaban perangkat dan juga di fungsikan untuk beberapa pengembangan lebih lanjut yang bertujuan menggantikan proses pemrograman manual yang membutuhkan keahlian khusus dan tidak semua orang yang mampu melakukannya.

Kata Kunci : Arduino,*Do It By Yourself*,Pengaplikasian *software*

-----Halaman ini sengaja dikosongkan-----

*GENERIC PROGRAMMABLE CONNECTING DESIGN USING I / O
FROM ARDUINO*

Nama : Rendra Kurnia .R

Nama : Setyo Budi Utomo

Pembimbing 1 : Ir. Djoko Suprajitno Rahardjo, M.T

Pembimbing 2 : Dr. Ir. Achmad Affandi, DEA

ABSTRACT

Programming the Arduino device today requires the ability to master the Arduino programming language itself, it makes ordinary people who want to assemble input-output sensor with difficulty to find a single-board micro controller that is open source with a cheap budget. So they must learn Arduino programming language to start from scratch and also in determining the sensor library must find library that matches the type of inputs and outputs to be connected to arduino and it was not efficient at all.

This Thesis is an implementation of the software that we create for enabling users to program the arduino do not have to learn English perograman again just use the software that we create and we named "Do It By Yourself 'which facilitate the programming process, only to determine the type of Arduino and the type of input and the output that has been provided then the software will automatically embed the library into arduino sensor in accordance with the specified input output was at once wrote a programming language.

In this software we only limited a few inputs and outputs only because of limited abbility support components, in this case our software is applied to program the security system process and monitoring the temperature and humidity of the device and also enabled for some further development that aims to replace the manual programming process Which requires special skills and not everyone is capable of doing so.

Keywords: *Arduino, Do It By Yourself, Application software*

-----Halaman ini sengaja dikosongkan-----

KATA PENGANTAR

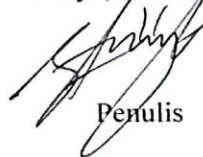
Puji syukur penulis panjatkan kehadirat Allah SWT yang selalu memberikan rahmat dan hidayah-Nya sehingga Tugas Akhir ini dapat terselesaikan dengan baik. Shalawat serta salam semoga selalu dilimpahkan kepada Rasulullah Muhammad SAW, keluarga, sahabat, dan umat muslim yang senantiasa meneladani beliau.

Tugas Akhir ini disusun untuk memenuhi sebagian persyaratan guna menyelesaikan pendidikan diploma pada Bidang Studi Komputer Kontrol, Program Studi D3 Teknik Elektro, Fakultas Teknologi Industri, Institut Teknologi Sepuluh Nopember Surabaya dengan judul: **RANCANG BANGUN GENERIC PROGRAMMABLE PENGHUBUNG PENGGUNA DENGAN I/O DARI ARDUINO**

Penulis mengucapkan terima kasih kepada Ibu dan Bapak penulis yang memberikan berbagai bentuk doa serta dukungan tulus tiada henti, Bapak Ir. Djoko Suprajitno Rahardjo, M.T., dan bapak Dr. Ir. Achmad Affandi, DEA. atas segala bimbingan ilmu, moral, dan spiritual dari awal hingga terselesaikannya Tugas Akhir ini, kedua orang tua yang selalu memberikan doa, semangat, dan dukungannya kepada penulis Penulis juga mengucapkan banyak terima kasih kepada semua pihak yang telah membantu baik secara langsung maupun tidak langsung dalam proses penyelesaian Tugas Akhir ini.

Penulis menyadari dan memohon maaf atas segala kekurangan pada Tugas Akhir ini. Akhir kata, semoga Tugas Akhir ini dapat bermanfaat dalam pengembangan keilmuan di kemudian hari.

Surabaya, 9 Juni 2017



Penulis

-----Halaman ini sengaja dikosongkan-----

DAFTAR ISI

	HALAMAN
HALAMAN JUDUL.....	i
HALAMAN JUDUL.....	i
PERNYATAAN KEASLIAN TUGAS AKHIR	v
HALAMAN PENGESAHAN	vii
ABSTRAK	ix
<i>ABSTRACT</i>	xi
KATA PENGANTAR.....	xiii
DAFTAR ISI.....	xv
DAFTAR GAMBAR	xvii
DAFTAR TABEL	xix
BAB I PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Permasalahan	2
1.3 Batasan Masalah.....	2
1.4 Tujuan	3
1.5 Metodologi Penelitian	4
1.6 Sistematika Laporan.....	7
1.7 Relevansi	8
BAB II TEORI DASAR	9
2.1 Tinjauan Pustaka	9
2.2 Microsoft Visual Basic 6.0.....	10
2.2.1 Menu Bar.....	12
2.2.2 Tittle Bar	13
2.2.3 Tool Bar	14
2.2.4 Toolbox	14
2.2.5 Project Windows	15
2.2.6 Properties Windows	15
2.2.7 Form Layout Windows.....	16
2.2.8 Form Window	16
2.2.9 Immediate Window	17
2.2.10 Code Window.....	17
2.2.11 Event	18

2.2.12 Method	18
2.2.13 Module	18
2.3 Interface	18
2.4 Komunikasi Serial Port Visual Basic 6.0 Dengan Arduino	18
2.5 Arduino Mega	20
2.6 Input Output Analog	29
BAB III PERENCANAAN DAN PEMBUATAN ALAT.....	33
3.1 Blok Fungsional Sistem	33
3.2 Kelebihan dan Kekurangan Komponen Pada <i>Software</i>	35
3.3 Flowchart Diagram	55
3.4 Pembagian Tugas Pekerjaan	56
3.4.1 Proses Pengerjaan Blok “B”	58
3.4.2 Proses Penggabungan Pekerjaan Blok C.....	82
BAB IV PENGUJIAN DAN ANALISA DATA.....	85
4.1 Pengujian Software	85
4.2 Pengujian Mengubungkan Software	94
BAB V KESIMPULAN DAN SARAN	97
5.1 Kesimpulan	97
5.2 Saran dan Masukan	97
DAFTAR PUSTAKA.....	99
LAMPIRAN A	101
A.1. Source Program Visual Basic	101
DAFTAR RIWAYAT HIDUP	139

DAFTAR GAMBAR

HALAMAN

Gambar 1.1 Metode The Classic Life Cycle / Waterfall	4
Gambar 1.2 Diagram Blok Pembagian Tugas Kelompok	6
Gambar 2.1 Tampilan Awal Visual Basic 6.0	12
Gambar 2.2 Tampilan Menu <i>Bar</i>	13
Gambar 2.3 Tampilan <i>Tittle Bar</i>	13
Gambar 2.4 Tampilan <i>Tool Bar</i>	14
Gambar 2.5 Tampilan <i>Toolbox</i>	14
Gambar 2.6 Tampilan <i>Project Windows</i>	15
Gambar 2.7 Tampilan <i>Properties Windows</i>	15
Gambar 2.8 Tampilan <i>Form Layout Windows</i>	16
Gambar 2.9 Tampilan <i>Form Windows</i>	16
Gambar 2.10 Tampilan <i>Immediate Window</i>	17
Gambar 2.11 Tampilan <i>Code Window</i>	17
Gambar 2.12 Pinout Arduino Mega 2560	22
Gambar 2.13 Tampilan <i>Sketch</i> di Arduino IDE	28
Gambar 2.14 Tampilan Sinyal <i>Input Analog</i>	30
Gambar 2.15 Tampilan Sinyal <i>Output Analog</i>	31
Gambar 2.16 Tampilan PWM Setelah Diberikan Filter	31
Gambar 2.17 Tampilan Sinyal <i>Input Digital</i>	32
Gambar 3.1 Pinout Arduino Mega 2560	33
Gambar 3.2 Sensor Fingerprint Optical SM630	37
Gambar 3.3 Spesifikasi Sensor Suhu dan Kelembaban SHT	41
Gambar 3.4 Pinout Sensor SHT11 dan DHT22	42
Gambar 3.5 Pinout Arduino Uno	45
Gambar 3.6 Pinout Arduino Mini	46
Gambar 3.7 Pinout Arduino Nano	47
Gambar 3.8 Karakter Pada LCD	49
Gambar 3.9 Gambar Visual Kipas Motor DC 5V	50
Gambar 3.10 <i>Flowchart</i> Proses Desain Kebutuhan <i>Platform</i>	55
Gambar 3.11 <i>Flowchart</i> Pembagian Tugas Kelompok	57
Gambar 3.12 Proses Pengerjaan Alur B	58
Gambar 3.13 <i>Flowchart</i> Perancangan <i>Software “Do It By Yourself”</i>	62

Gambar 3.14 Langkah-langkah Membuat <i>New Project</i>	63
Gambar 3.15 <i>Form</i> memilih <i>Port</i> Arduino	63
Gambar 3.16 Petunjuk Mengisikan Kode Pemrograman Pada <i>Form</i>	64
Gambar 3.17 Membuat <i>Form</i> Untuk Memilih Sensor	68
Gambar 3.18 Membuat <i>Form</i> Memilih Tipe Arduino	72
Gambar 3.19 Membuat <i>Form</i> Untuk memilih <i>Output</i>	75
Gambar 3.20 Membuat <i>Form</i> Melihat <i>DataSheet</i> Arduino.....	76
Gambar 3.21 Membuat <i>Form</i> Program Yang Akan Diunggah ke Arduino.....	80
Gambar 4.1 Tampilan Jendela Pemilihan Port.....	85
Gambar 4.2 Tampilan Jendela Saat Port Arduino Terdeteksi	85
Gambar 4.3 Tampilan Jendela Pemilihan Sensor Yang Akan Dipakai..	86
Gambar 4.4 Tampilan Jendela <i>Link Data Sheet</i> Sensor Yang Dipilih....	86
Gambar 4.5 Tampilan Jendela Apabila Sensor Belum Dipilih	87
Gambar 4.6 Tampilan Jendela Pemilihan Tipe Arduino.....	87
Gambar 4.7 Jendela Peringatan Tidak Memilih Tipe Arduino	88
Gambar 4.8 Tampilan Jendela Pemilihan Jenis Sensor <i>Output</i>	88
Gambar 4.9 Jendela Peringatan Tidak Memilih Jenis <i>Output</i>	89
Gambar 4.10 Tampilan Jendela Konfigurasi	89
Gambar 4.11 Tampilan Jendela <i>Library</i> Yang Dibutuhkan.....	90
Gambar 4.12 Tampilan Jendela Petunjuk Cara Menginstal <i>Library</i>	90
Gambar 4.13 Tampilan Jendela Bantuan & <i>Contact Person</i>	91
Gambar 4.14 Tampilan Jendela Cara Pemasangan Program	91
Gambar 4.15 Tampilan Jendela Program Utama	92
Gambar 4.16 Tampilan Jendela Program <i>Enroll</i>	92
Gambar 4.17 Tampilan Jendela Pengaturan Pin	93
Gambar 4.18 Tampilan Jendela Pemberitahuan Proses Telah Selesai ...	93
Gambar 4.19 Tampilan Jendela Pemberitahuan Port Terdeteksi	94
Gambar 4.20 Tampilan Jendela Pemilihan Port Berhasil	94
Gambar 4.21 Tampilan Jendela Pemilihan Port Berhasil	95

DAFTAR TABEL

	HALAMAN
Tabel 2.1 Spesifikasi Dari Arduino Mega 2560	22
Tabel 2.2 Tabel Pin Serial RX dan TX	24
Tabel 2.3 Tabel Pin Eksternal Interupsi	25
Tabel 2.4 Tabel Pin <i>SPI</i>	25
Tabel 3.1 Spesifikasi Dari Tipe-Tipe Arduino	43

-----Halaman ini sengaja dikosongkan-----

BAB I

PENDAHULUAN

1.1 Latar Belakang

Seiring dengan berkembangnya dunia teknologi komputer yang sangat pesat, banyak dari perusahaan dan instansi pemerintahan yang berlomba – lomba merancang sebuah aplikasi yang mudah dalam penggunaannya, baik dari segi perancangan antarmuka (*interface*) maupun segi teknis lainnya. Hal ini dimaksudkan agar *user* yang menggunakan aplikasi tidak cepat bosan dan dapat dengan cepat memahami maksud dari tata letak berbagai macam navigasi yang ada pada aplikasi.

Arduino adalah salah satu *mikrokontroler* yang banyak di gunakan oleh kebanyakan orang dan harganya yang terjangkau yang merupakan miniatur dari PLC yang tidak terlalu rumit dalam pengembangannya. Permasalahan yang selalu timbul pad Arduino adalah kesulitan user dalam melakukan proses pengolahan data dan juga dalam proses pemrogramannya yang rumit karena bahasa pemrogramannya membutuhkan orang yang mengerti akan bahasa pemrograman dasar. Selain proses yang dilakukan masih secara manual, hal lain yang menjadi permasalahan adalah banyaknya aplikasi – aplikasi dengan desain *interface* yang membuat *user* merasa asing dalam menggunakan aplikasi. Sehingga *user* terkadang membutuhkan waktu yang lebih lama untuk beradaptasi dengan aplikasi tersebut. Hal ini disebabkan karena desain interface aplikasi terlihat buruk sehingga membingungkan user bahkan bisa membuat *user* frustasi dan bosan dalam menggunakan aplikasi penghubung tersebut, dan juga banyaknya program dan *library* sensor yang beredar di internet yang belum tentu cocok dengan tipe sensor *input* atau *output* yang telah *user* punyai sehingga harus memilih satu persatu dan itu sangat tidak efisien.

Faktor lain yang membuat *user* selalu merasa kesulitan adalah tidak adanya perbedaan antara tampilan untuk tipe *user* yang sudah *expert* dan tampilan untuk tipe *user* yang masih pemula.

Melalui permasalahan yang timbul, maka di harapkan adanya perancangan aplikasi *user interface* dengan desain *interface* yang mudah dipahami oleh *user*, baik itu dari tampilan, struktur menu, navigasi – navigasi, pemilihan kontrol yang tepat bagi *user* dan aliran sistem yang

sesuai dengan tugas *user*, dengan rancangan pemilihan *input* dan *output*, serta pemilihan tipe Arduino, beserta program dan *library* yang telah di saring agar cocok dengan tipe sensor yang umum di gunakan.

Berdasarkan penjelasan latar belakang di atas, maka penulis memberikan solusi untuk memulai tahap perancangan desain *interface* pada Arduino dengan paket program dan *library* yang telah di seleksi dengan judul **“Rancang Bangun Generic Programmable Penghubung Pengguna Dengan I/O Dari Arduino”**.

Yang memiliki beberapa kelebihan yang tentunya sangat dibutuhkan seperti: layak (*feasible*), berbeda satu sama lain (*distinct*), tetap (*penent*), akurat (*accurate*), handal (*reliable*) dan dapat diterima (*acceptable*).

1.2 Permasalahan

Yang menjadi rumusan masalah pada rancangan Alat Tugas Akhir ini ada diantaranya :

- Dibutuhkan *input* maupun *output* atau *library* untuk Arduino yang secara tidak menentu karena disesuaikan dengan kondisi atau permintaan dari *user* atau orang – orang awam.
- Membuat *database* dari keseluruhan program untuk *input* dan *ouput* yang biasanya sebagian besar banyak yang digunakan atau dibutuhkan meskipun ada beberapa yang tidak digunakan atau hanya beberapa user yang membutuhkannya.
- Bagaimana merancang sebuah desain *interface* yang baik agar user tidak cepat bosan dalam menggunakan aplikasi pemrograman Arduino.
- Bagaimana melakukan perancangan desain *interface* yang mudah dipahami oleh user yang bertipe expert dan pemula.
- Bagaimana menciptakan suatu aplikasi yang menggantikan tugas *user* dalam mengetik bahasa pemrograman Arduino sekaligus menanamkan *library* yang cocok untuk jenis sensor *input/output* serta tipe Arduino yang berbeda-beda.

1.3 Batasan Masalah

Dalam perancangan software Tugas Akhir ini, adapun beberapa masalah yang memiliki batasan untuk memfokuskan pada pembuatan software pada Tugas Akhir ini yang diantaranya.

- *Plant* yang dibuat adalah alat ukur/sensor yang umum di pasaran dan terjangkau dengan dana pengadaan alat.
- Aplikasi yang dirancang lebih dipusatkan pada konsep dan karakteristik perancangan desain *interface software* “*Do It By Yourself*”.
- Ukuran Layar/*Screen Layout* yang konsisten.
- Pesan/*Message* pada sebuah validasi dibuat lebih singkat dan jelas.
- Penggunaan suatu perintah dengan menggunakan *Combo Box*, & *Button*.
- *Tools* yang digunakan untuk merancang aplikasi ini antara lain *Microsoft Visual Basic 2010*.

1.4 Tujuan

Dalam hal ini, tujuan untuk rancangan dari alat Tugas Akhir ini untuk bermaksud meninjau masalah – masalah yang mudah dan tentunya sudah umum dikalangan masyarakat terutama kalangan masyarakat awam yang belum sepenuhnya mengetahui kemajuan dari teknologi yang ada pada saat ini. Maka dari itu, tujuan untuk adanya rancangan alat Tugas Akhir ini untuk mempermudah suatu pekerjaan yang sebelumnya sudah umum berada dikalangan masyarakat yang dimana biasanya dilakukan secara manual akan di ubah menjadi otomatis sehingga lebih efisien dan praktis. Dan juga rancangan *software* ini dapat membantu masyarakat atau orang – orang awam yang masih tidak mengerti tentang cara kerja maupun kebutuhan *hardware* dan *software* yang diperlukan untuk mengatasi masalah yang dibutuhkan oleh *user* serta rancang bangun *software* ini juga mencakup untuk membantu permasalahan tersebut sehingga hal – hal yang berhubungan dengan hal – hal yang biasanya dilakukan secara manual yang tentunya membuang – buang waktu, monoton, kerugian fisik maupun non – fisik akan sedikit teratasi secara cepat dan tepat.

Adapun tujuan dari perancangan Tugas Akhir ini ialah :

- Pembuatan *software* yang berisikan kumpulan *database* program sebagai *interface* antara *user* dengan perangkat *hardware* yang disesuaikan untuk *input* maupun *output* yang dibutuhkan oleh *user*.
- Memberikan kemudahan bagi semua tipe *user* dalam menggunakan aplikasi, tanpa membedakan desain *interface* untuk *expert* ataupun pemula.

- Membantu pengguna Arduino dalam merancang sebuah aplikasi yang menerapkan prinsip desain *interface* untuk menghasilkan tampilan yang berkualitas.

1.5 Metodologi Penelitian

Dalam rancangan untuk Tugas Akhir ini yang berupa inovasi dari alat yang sudah ada namun mengotomatisasi yang mengacu pada 3 aspek yakni sektor keamanan, sektor otomatisasi serta sektor monitoring suhu dan kelembaban.

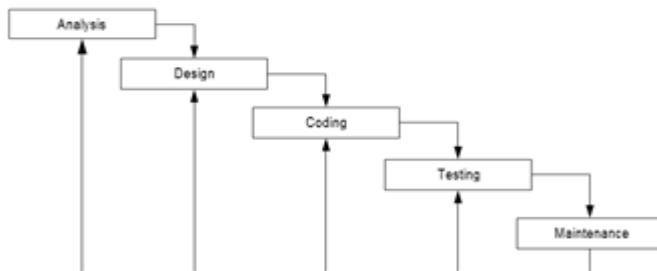
a. Studi Pustaka dan Survey Data Awal :

Pada tahap ini akan di lakukan studi pustaka berkaitan :

- 1) Artikel – artikel yang berkaitan dengan konfigurasi pemrograman komponen dan sensor dari *input* dan *output* yang akan digunakan dalam proses otomatisasi.
- 2) Karakteristik *library* masing – masing komponen dan sensor dari *input* dan *output*.
- 3) Membuat rancangan *interface* yang simpel agar pengaturan mudah dalam mengatur sensor dari *input* dan *output*.
- 4) Penggunaan arduino sebagai pusat pengendali *input* dan *output*.

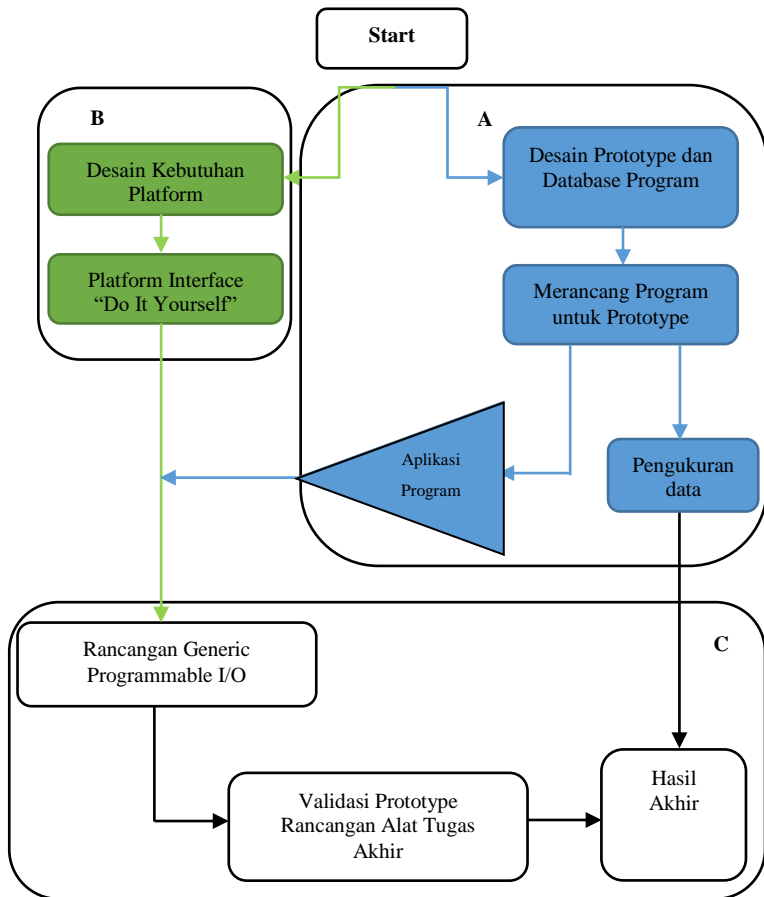
b. Perencanaan dan Pembuatan Software :

Dalam pengembangan aplikasi ini digunakan metode The Classic Life Cycle (Paradigma Waterfall). Pada metode ini terdapat 5 tahap untuk mengembangkan suatu perangkat lunak. Kelima tahapan itu tersusun dari atas kebawah, diantaranya : Analysis, Design, Coding, Testing, Maintenance. Tahap- tahap pengembangan perangkat lunak metode waterfall dapat dilihat pada gambar 1.1.



Gambar 1.1 Metode The Classic Life Cycle / Waterfall

- a. Analysis adalah tahap menganalisa hal-hal yang diperlukan dalam pelaksanaan proyek pembuatan atau pengembangan software.
- b. Design adalah tahap penerjemah dari keperluan-keperluan yang dianalisis ke dalam bentuk yang lebih mudah dimengerti oleh pemakai. Yaitu dengan cara menampilkan ke dalam Diagram Konteks, Data Flow Diagram (Diagram Aliran Data), Entity Relationship Diagram, Struktur Tabel, dan Struktur Menu.
- c. Coding adalah tahap penerjemah data/pemecahan masalah software yang telah dirancang ke dalam bahasa pemrograman yang telah ditentukan.
- d. Testing adalah tahap pengujian terhadap program yang telah dibuat. Pengujian ini dimulai dengan membuat suatu uji kasus untuk setiap fungsi pada perangkat lunak, kemudian dilanjutkan dengan pengujian terhadap modul-modul dan terakhir pada tampilan antar muka untuk memastikan tidak ada kesalahan dan semua berjalan dengan baik dan input yang diberikan hasilnya sesuai dengan yang diinginkan.
- e. Maintenance adalah perangkat lunak yang telah dibuat dapat mengalami perubahan sesuai permintaan pemakai. Pemeliharaan dapat dilakukan jika ada permintaan tambahan fungsi sesuai dengan keinginan pemakai ataupun adanya pertumbuhan dan perkembangan baik perangkat lunak maupun perangkat keras.



Gambar 1.2 Diagram Blok Pembagian Tugas Kelompok

Keterangan :

Pada gambar 1.2 Blok “B” yaitu desain kebutuhan *platform* yang dimaksudkan yaitu desain *hardware* yang dimana dapat menjalankan sistem atau program dapat berjalan sesuai dengan rancangan. Selanjutnya *platform interface* “*Do It by Yourself*” dimana dari hal ini dibutuhkan sistem yang dapat dijalankan sesuai dengan kebutuhan *user*

tanpa harus mengetahui secara detail bagaimana alur dari jalannya program yang dibutuhkan.

Lalu dari kedua bagian tersebut digabungkan dan menjadi sebuah *Generic Programmable I/O*. Dimana pada bagian ini rancangan alat yang masih berupa secara “kasar” karena masih uji coba dengan data data yang sebelumnya diambil.

Selanjutnya ada validasi *prototype* rancangan *software* Tugas Akhir yang dimana hasil evaluasi dari proses sebelumnya agar rancangan alat lebih sempurna dan mempunyai data yang tepat sesuai dengan kebutuhan *user* dari rancangan sistem.

c. Perencanaan dan Pembuatan *Generic Programmable*:

Pada tahap ini akan di lakukan perancangan program (*software*) sesuai dengan data yang telah di dapat dan dipelajari dari studi pustaka, dan kemudian program tersebut di terapkan dan diupload ke Arduino.

d. Uji Coba dan Analisis Data:

Pada tahap ini akan di lakukan pengujian program yang dibutuhkan oleh *user* yang telah diambil dari *database*. Selanjutnya akan di lakukan juga pengukuran ketepatan *input* dan *output* berdasarkan data sementara yang telah diambil.

- 1) Uji coba dilakukan dengan melakukan pengujian penghubungan *software* dengan Arduino agar data yang didapatkan memiliki hasil yang sesuai dengan perhitungan atau perencanaan sebelumnya.
- 2) Uji coba atau analisa data dari *input* dan *output* yang diterapkan pada *prototype* sebelum diaplikasikan ke *platform* agar tidak terjadi kesalahan pada saat penerapan.

e. Penyusunan Laporan

Pada tahap ini akan di lakukan penyusunan laporan hasil dari pembuatan rancangan desain *software* dengan beberapa tipe *input output* yang sesuai dengan *platform* yang telah didesain dan dibutuhkan oleh *user* dan hasil akhir dari rancangan tersebut adalah *Generic Programmable* yang sesuai dengan rencana yang telah di buat.

1.6 Sistematika Laporan

Pembahasan Tugas Akhir ini akan dibagi menjadi lima Bab dengan sistematika sebagai berikut :

Bab I Pendahuluan

Pada bab ini berisi tentang latar belakang, maksud dan tujuan, identifikasi masalah, batasan masalah, metode penelitian, dan sistematika laporan.

Bab II Teori Dasar

Bab ini menjelaskan tentang tinjauan pustaka, konsep dari Microsoft *Visual Basic 6.0*, *Interface*, Arduino Mega, Komunikasi serial port VB dengan Arduino, *Input Output* analog dan digital Arduino.

Bab III Perancangan Sistem

Pada bab ini berisi tentang deskripsi dari perancangan perangkat lunak.

Bab IV Simulasi, Implementasi dan Analisis Sistem

Bab ini memuat hasil simulasi dan implementasi serta analisis data dari hasil tersebut.

Bab V Penutup

Bab ini berisi kesimpulan dan saran dari hasil pembahasan yang telah diperoleh.

1.7 Relevansi

Manfaat dari Tugas Akhir ini adalah membantu dan memudahkan masyarakat yang di bidang teknik elektro maupun orang awam yang bukan dari bidang elektro dalam menyikapi dan mengatasi masalah – masalah umum yang sering terjadi dan menyesuaikan dengan kemajuan atau perkembangan teknologi yang ada tanpa harus mempelajari atau mengetahuinya secara jelas dan rinci.

BAB II

TEORI DASAR

2.1 Tinjauan Pustaka

Pada pembuatan skripsi ini, penulis menggunakan teori-teori yang bermanfaat dalam perancangan aplikasi diantaranya adalah pengertian *artificial intelligence* atau kecerdasan buatan, pengertian *expert system* atau sistem pakar, metode forward-chaining, sistem operasi Android, dan hal-hal yang terkait dalam penulisan skripsi ini.

Definisi Perangkat Lunak Menurut Pressman (2010, p4), definisi dari perangkat lunak dapat diartikan sebagai berikut:

1. Instruksi (program komputer) saat dieksekusi dapat menyediakan *features*, fungsi, dan performa yang sesuai dan diinginkan oleh pengguna.
2. Struktur data yang memungkinkan sebuah program dapat memanipulasi informasi dengan baik.
3. Informasi deskriptif berupa *hard copy* dan bentuk *virtual* yang dapat menggambarkan operasi dan kegunaan dari suatu program.

Sedangkan Karakteristik Perangkat Lunak diantaranya Karakteristik perangkat lunak menurut Pressman 2010, p4), jauh berbeda bila dibandingkan dengan perangkat keras. Karakteristik yang dimiliki perangkat lunak sebagai berikut:

1. Perangkat lunak dapat dikembangkan dan direkayasa, tidak di produksi dalam pengertian klasik.
2. Perangkat lunak tidak pernah kadaluwarsa.
3. Meskipun industri-industri melakukan perakitan berbasis komponen, hampir semua perangkat lunak tetap dibangun secara tradisional.

Rekayasa Perangkat Lunak Definisi Rekayasa Perangkat Lunak Menurut Fritz Bauer [NAU69], rekayasa perangkat lunak adalah pembuatan dan penggunaan prinsip-prinsip rekayasa untuk memperoleh perangkat lunak yang ekonomis dan dapat bekerja secara efisien dan handal pada suatu komputer. IEEE [IEE93a] mengembangkan definisi rekayasa perangkat lunak menjadi lebih komprehensif, definisi rekayasa perangkat lunak adalah sebuah studi dan aplikasi dari sebuah penerapan kuantitatif secara sistematis, disiplin, dan sistematis terhadap

pengembangan, operasi dan pemeliharaan perangkat lunak yang merupakan aplikasi rekayasa berkaitan dengan semua perangkat lunak.

Pressman (2010, p14) memberikan definisi rekayasa perangkat lunak menjadi beberapa lapisan yang terdiri dari:

1. Fokus pada kualitas (*A quality focus*)

Setiap pendekatan rekayasa (termasuk rekayasa perangkat lunak) harus memiliki komitmen terhadap kualitas produk yang dihasilkan. Budaya inilah yang dapat mendukung rekayasa perangkat lunak adalah fokus pada kualitas.

2. Proses (*Process*)

Dasar dari rekayasa perangkat lunak adalah lapisan proses. Proses mendefinisikan *framework* untuk suatu *Key Process Areas* (KPA's) yang harus dibentuk untuk melakukan pengiriman teknologi rekayasa perangkat lunak agar berjalan efektif.

3. Metode (*Method*)

Pada lapisan metode menerangkan secara teknis bagaimana cara membangun perangkat lunak. Metode ini memiliki tahapan yang mencakup analisis kebutuhan (*requirement analysis*), desain (*design*), pembuatan program (*program construction*), pengujian (*testing*), dan pemeliharaan perangkat lunak (*maintenance*).

4. Alat (*Tools*)

Alat dalam rekayasa perangkat lunak memberikan otomatis dan semi-otomatis dalam proses dan metode. Sebuah sistem untuk mendukung dalam pengembangan perangkat lunak disebut *Computer-Aided Software Engineering* (CASE). Case dapat menggabungkan perangkat lunak, perangkat keras dan database untuk membangun lingkungan rekayasa perangkat lunak dengan CAD/CAE (*Computer-Aided Design/ Engineering*) untuk perangkat keras.

2.2 Microsoft Visual Basic 6.0

Visual Basic merupakan salah satu program yang berdasarkan GUI (*Graphical User Interface*), sehingga tidak perlu membuat instruksi pemrograman dengan kode baris, cukup dengan pembuatan program yang berbasis objek. Visual Basic juga didukung oleh Bahasa Basic yang dikenal mudah. Bahasa Basic merupakan bahasa yang mengandung pernyataan, fungsi dan *keyword* yang langsung terhubung dengan windows. Microsoft Visual Basic 6.0 adalah bahasa program yang bekerja dalam lingkup Microsoft Windows, Visual

Basic berasal dari bahasa pemrograman yang populer disebut Basic (*Beginner's All Purpose Symbol Instruction Code*). Bahasa basic diperkenalkan pertama kali oleh *Dartmouth College* pada tahun 1960. Kemudian beberapa pengembang *software* mempopulerkan Bahasa Basic dengan membuat berbagai aplikasi seperti GW-Basic, Qbasic dan *Quick Basic*. Visual Basic berbasis Windows diperkenalkan pada tahun 1991. [4]

Visual Basic memiliki kelebihan-kelebihan yang tidak dimiliki oleh versi sebelumnya, kelebihanannya antara lain *compiler* (proses *compile*) dapat dilakukan dengan cepat, mendukung control data objek yang baru, mendukung berbagai macam database, pembuatan laporan yang lebih mudah dan mendukung pengaksesan terhadap internet. Visual Basic 6.0 menyediakan tiga macam *interface* yang bisa digunakan untuk merancang aplikasi sesuai dengan kebutuhan. Interface tersebut berupa MDI (*Multi Document Interface*), SDI (*Single Document Interface*), dan EDI (*Explorer Document Interface*).

Visual Basic 6 merupakan salah satu *tool* untuk pengembangan aplikasi yang banyak diminati oleh orang. Di sini Visual Basic 6 menawarkan kemudahan dalam pembuatan aplikasi dan dapat menggunakan komponen-komponen yang telah disediakan. Untuk memulai Visual Basic 6 anda perlu menginstall Visual Basic 6.0. Program ini biasanya dalam satu paket dengan Visual Studio 6.0. Dengan menggunakan Visual Basic 6 kita bisa menghasilkan berbagai macam jenis program. Dari aplikasi yang mengintegrasikan database, jaringan, office *automation*, dan *web application*. Di sini kita akan membahas dasar-dasar pemrograman Visual Basic dengan sedikit tentang database.



Gambar 2.1 Tampilan Awal Visual Basic 6.0

Pada gambar 2.1 tampilan layar awal akan muncul tampilan seperti di atas. Visual Basic menyediakan banyak jenis modul aplikasi. Untuk memulai program standar pilihlah Standard EXE, kemudian klik pada tombol *Open*. Setelah itu akan muncul tampilan seperti berikut ini. Sekarang kita akan mengenal bagian-bagian dari IDE (*Integrated Development Environment*) yang kita gunakan. Mengetahui lingkungan kerja Visual Basic adalah bagian awal yang harus dikerjakan untuk memulai membuat program. Bagian penting yang perlu diketahui setelah program visual basic dijalankan antara lain adalah komponen-komponen Visual Basic 6.0.[4]

2.2.1 Menu Bar

Menu bar berfungsi untuk menjalankan suatu perintah atau fungsi pengembangan project Visual Basic. Menu ini dikelompokkan kedalam kelompok-kelompok tertentu yang akan membantu kita melakukan design aplikasi. Berikut penjelasan mengenai pengelompokan-pengelompokan menu-menu tersebut seperti gambar 2.2.

- *File* : Terdiri dari perintah-perintah untuk membuka, menutup, menyimpan, mencetak, mengcompile *project* yang sedang kita kerjakan.
- *Edit* : Kumpulan perintah yang membantu kita memanipulasi penulisan kode (*listing* program).
- *View* : Terdiri dari perintah untuk melihat/berpindah antar *window* didalam lingkungan pengembangan.
- *Project* : Sekumpulan perintah untuk menambah komponen *Project* (*Form*, *Module*, dll) dan tempat dimana kita akan men *setup properties project*.
- *Format* : Perintah- perintah untuk memanipulasi kontrol yang terdapat pada sebuah *Form*. [5]



Gambar 2.2 Tampilan Menu Bar

2.2.2 Tittle Bar

Tittle bar berfungsi untuk menampilkan judul proyek Visual Basic yang sedang dikerjakan seperti gambar 2.3. [5]



Gambar 2.3 Tampilan Tittle Bar

2.2.3 Tool Bar

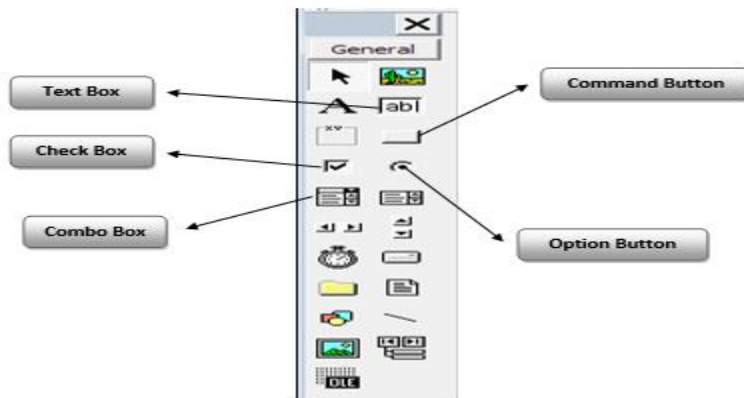
Seperti gambar 2.4 *tool bar* berisi *icon-icon* yang mewakili perintah-perintah yang terdapat di dalam *Menu bar* (*Shortcut*). *Toolbar* ini secara default berisi perintah-perintah umum untuk memanipulasi *project*. *Toolbar* bisa diset sesuai dengan keinginan kita sendiri. [5]



Gambar 2.4 Tampilan *Tool Bar*

2.2.4 Toolbox

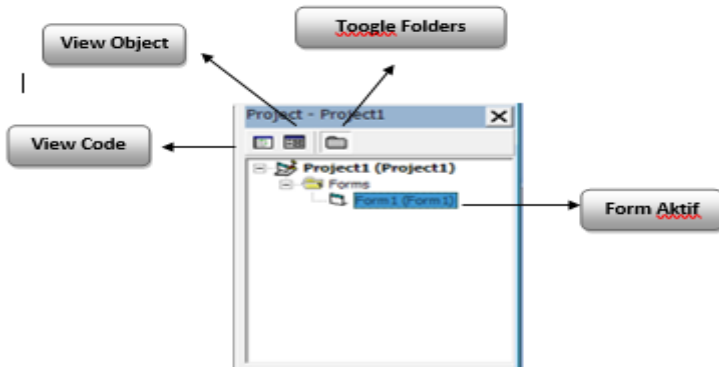
Berisi kontrol-kontrol yang dapat kita pakai /letakkan di dalam sebuah *Form*. Secara *default* berisi kontrol-kontrol umum yang sudah disiapkan oleh Visual Basic. Kita dapat menambahkan sendiri kontrol-kontrol yang kita inginkan seperti gambar 2.5. (klik kanan pada area *Tool Box* dan pilih *Components*).



Gambar 2.5 Tampilan *Toolbox*

2.2.5 Project Windows

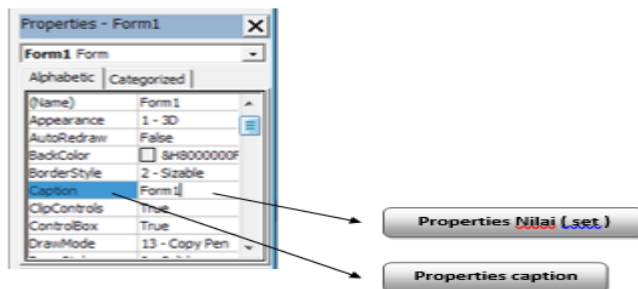
Project Window seperti gambar 2.6 berisi semua komponen yang terdapat didalam *project* yang sedang kita kembangkan (*Form, Module, Class, dll*).



Gambar 2.6 Tampilan *Project Windows*

2.2.6 Properties Windows

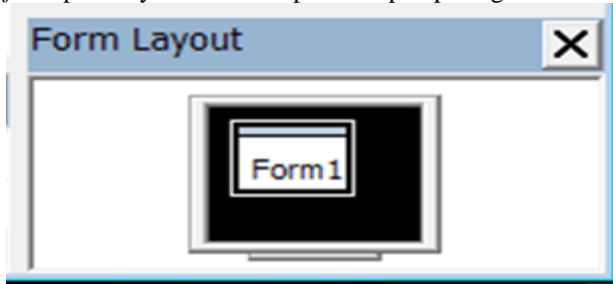
Properties Windows akan memperlihatkan semua *property* yang tersedia pada sebuah *object/control/component* yang sedang aktif (*focus*). Kita dapat melakukan manipulasi *properties* terhadap objek-objek tersebut seperti gambar 2.7. Secara *default* property-property itu diurutkan berdasarkan abjad. [5]



Gambar 2.7 Tampilan *Properties Windows*

2.2.7 Form Layout Windows

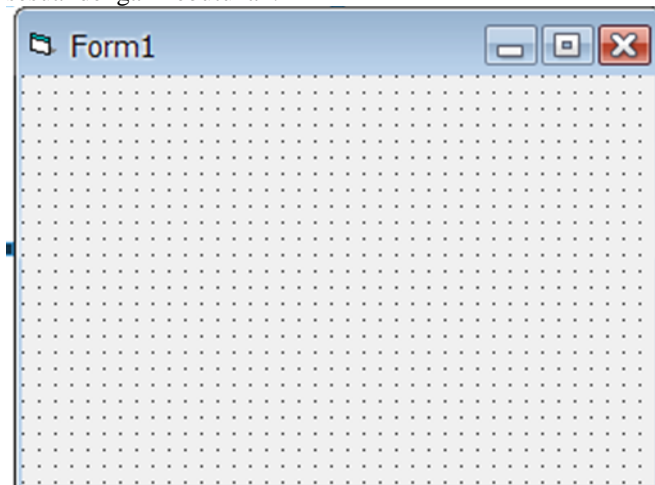
Form Layout Window digunakan untuk mengatur letak *form* pada layar monitor seperti tampak pada gambar 2.8.[5]



Gambar 2.8 Tampilan *Form Layout Windows*

2.2.8 Form Window

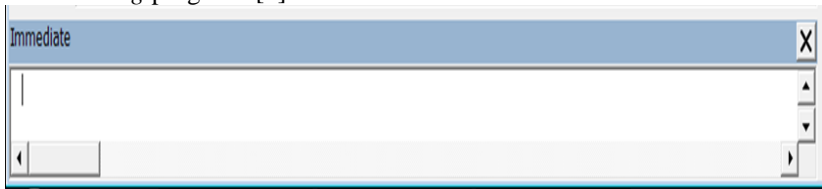
Form Window adalah lembar desain tampilan dari program yang dibuat seperti gambar 2.9. *Form* ini menjadi pondasi tempat diletakkannya kontrol-kontrol yang dimiliki oleh Visual Basic sesuai dengan yang diinginkan. Kita dapat menambahkan sebanyak mungkin *form* kedalam aplikasi kita sesuai dengan kebutuhan.



Gambar 2.9 Tampilan *Form Windows*

2.2.9 Immediate Window

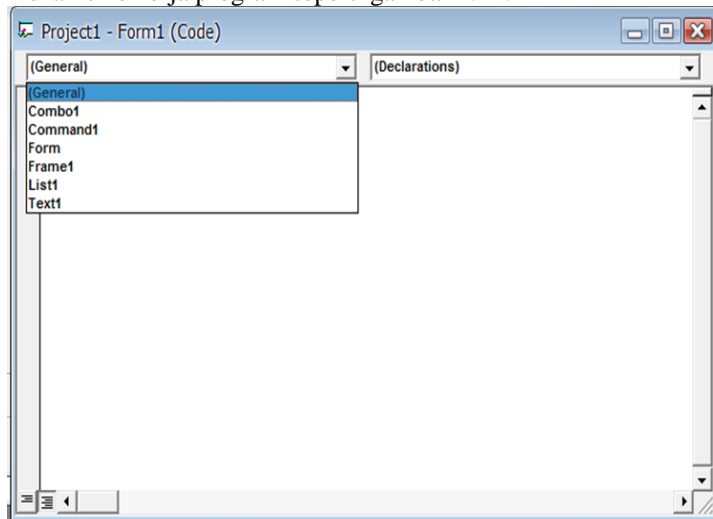
Immediate Window atau lebih dikenal dengan *debug window* adalah tempat dimana kita dapat melihat jalannya program secara step by step jika break mode dijalankan. Di dalam *window* ini seperti terlihat dalam gambar 2.10 kita dapat melakukan perhitungan aritmatika, melihat isi variable dan bahkan memanggil sebuah *procedure* yang terdapat di dalam *listing* program. [5]



Gambar 2.10 Tampilan *Immediate Window*

2.2.10 Code Window

Code Window adalah suatu jendela dimana merupakan tempat untuk menulis dan menyunting ruitn yang menentukan mekanisme kerja program seperti gambar 2.11.



Gambar 2.11 Tampilan *Code Window*

2.2.11 Event

Event adalah aksi yang dikenali oleh suatu objek, seperti klik pada *mouse* atau penekanan tombol, sehingga anda dapat menulis kode untuk menangani *event* tersebut.

2.2.12 Method

Method adalah suatu yang dapat dilakukan oleh objek atau kemampuan yang dimiliki oleh objek.

2.2.13 Module

Pada aplikasi tradisional sebuah prosedur telah diatur sendiri oleh aplikasi-aplikasinya. Proses ini merupakan bagian kode yang akan dieksekusi dan biasanya pengekseskuan dimulai dari baris pertama.

2.3 Interface

Desain *interface* adalah suatu proses interaktif yang melalui persyaratan diterjemahkan ke dalam “*blue print*” untuk membangun suatu perangkat lunak. Desain *interface* akan berfokus pada tiga area, yaitu : [4]

- 1.Desain *interface* antara modul –modul perangkat lunak.
- 2.Desain *interface* antara perangkat lunak dan prosedur dan konsumen informasi bukan manusia lainnya.
- 3.Desain *interface* antara seorang manusia (*user*) dan komputer.

Desain *interface* sendiri di bagi kedalam beberapa model di antaranya sebagai berikut ialah:

- a. Model desain dari keseluruhan sistem menggabungkan data, arsitektur,*interface* dan representasi perangkat lunak.
- b. Model pemakai menggambarkan *profile* para pemakai akhir dari sistem.
- c. Persepsi sistem merupakan citra yang ada di kepala seorang pemakai akhir(*client*).
- d. Citra sistem merangkai *manifestasi* bagian luar sistem berbasis komputer.

2.4 Komunikasi Serial Port Visual Basic 6.0 Dengan Arduino

Menghubungkan mikrokontroler Arduino dengan aplikasi *desktop* menggunakan Visual Basic (.NET), melalui komunikasi serial

(*serial port communication*). Perangkat utama yang saya gunakan adalah papan Arduino Mega dan Visual Basic 6.0. Kita akan membuat program di kedua sisi, yaitu antara mikrokontroler dan komputer akan saling berkomunikasi, jika *CommandButton* di aplikasi komputer kita klik, maka lampu pada Arduino akan menyala, demikian juga sebaliknya, jika salah satu port di Arduino kita tekan, maka akan muncul informasi di layar komputer. Pemrograman serial port pada Arduino Hubungkan Arduino ke PC Anda menggunakan kabel USB, lalu buka *software* Arduino. Program ini untuk memastikan bahwa koneksi antara Arduino dan PC tidak ada kendala. Ketik program seperti di bawah ini lalu upload ke Arduino. [2]

```
int inByte=0; // variable to receive data from serialport
void setup()
{ Serial.begin(9600);
}

void loop()
{
if (Serial.available() > 0)
{
inByte=Serial.read(); Serial.write(inByte);
}
}
```

Setelah itu buka *Tools > Serial Monitor* pada aplikasi Arduino, untuk melihat interaksi antara PC dan mikrokontroler. Jika kita mengirim sembarang data, maka Arduino akan membalas data yang sama. Jangan lupa periksa kecepatan komunikasi, keduanya harus sama. Dalam hal ini 9600 bit per detik.

Pada bagian *setup*, kita mengaktifkan fungsi serial port pada Arduino (yang terkoneksi langsung dengan port USB) dengan perintah:

- *Serial.begin(9600);* Perintah ini hanya dilakukan sekali saja, pada saat Arduino diaktifkan atau di-*reset*. Jika kita ingin kecepatan lebih tinggi, ganti dengan 115000. Jika kita ingin menggunakan serial port yang lain, gunakan *Serial1*, *Serial2* atau *Serial3*. (berlaku untuk mikro yang serial portnya lebih dari satu, contohnya Arduino Mega, terletak pada pin 14 sampai 19). Jika Anda menggunakan Uno, maka tidak ada pilihan lain.

- Fungsi `Serial.available()` memberikan nilai BENAR jika ada data yang masuk.
- Fungsi `Serial.read()` untuk membaca data yang masuk sebesar 1 byte. Perintah untuk mengirim data dari mikro ke luar adalah `Serial.write (data_yg_ditulis);`.

Dengan melihat program di atas, maka mikrokontroler hanya akan mengirim ke luar jika ada masukan data serial, dan pengirimannya akan diulang terus sebanyak berapa byte data yang masuk. Catatan mengenai komunikasi serial adalah pada contoh pengiriman data pada program ini, kita menggunakan karakter dengan nomor ASCII di bawah 128. Jika Anda ingin menerapkan pengiriman data yang lebih kompleks, dan membutuhkan karakter dengan nomor ASCII sampai 255, maka kita harus mengkonfigurasi *SerialPort* dengan menambahkan kode berikut di atas perintah *SerialPort.Open()*.

2.5 Arduino Mega

Microcontroller adalah sebuah sistem komputer fungsional dalam sebuah chip. Di dalamnya terkandung sebuah inti prosesor, memori (sejumlah kecil RAM, memori program, atau keduanya), dan perlengkapan *input* dan *output*. Dengan kata lain, *Microcontroller* adalah suatu alat elektronika digital yang mempunyai masukan dan keluaran serta kendali dengan program yang bisa ditulis dan dihapus dengan cara khusus, cara kerja *microcontroller* sebenarnya membaca dan menulis data. Sekedar contoh, bayangkan diri Anda saat mulai belajar membaca dan menulis, ketika Anda sudah bisa melakukan hal itu Anda bisa membaca tulisan apapun baik buku, cerpen, artikel dan sebagainya, dan Andapun bisa pula menulis hal – hal sebaliknya. Begitu pula jika Anda sudah mahir membaca dan menulis data maka Anda dapat membuat program untuk membuat suatu sistem pengaturan otomatis menggunakan *microcontroller* sesuai dengan keinginan Anda. *Microcontroller* merupakan komputer didalam chip yang digunakan untuk mengontrol peralatan elektronik, yang menekankan efisiensi dan efektifitas biaya. Secara harfiahnya bisa disebut “pengendali kecil” dimana sebuah sistem elektronik yang sebelumnya banyak memerlukan komponen – komponen pendukung seperti IC TTL dan CMOS dapat direduksi / diperkecil dan akhirnya terpusat serta dikendalikan oleh *microcontroller* ini. [2]

Microcontroller digunakan dalam produk dan alat yang dikendalikan secara otomatis, seperti sistem kontrol mesin, *remote*

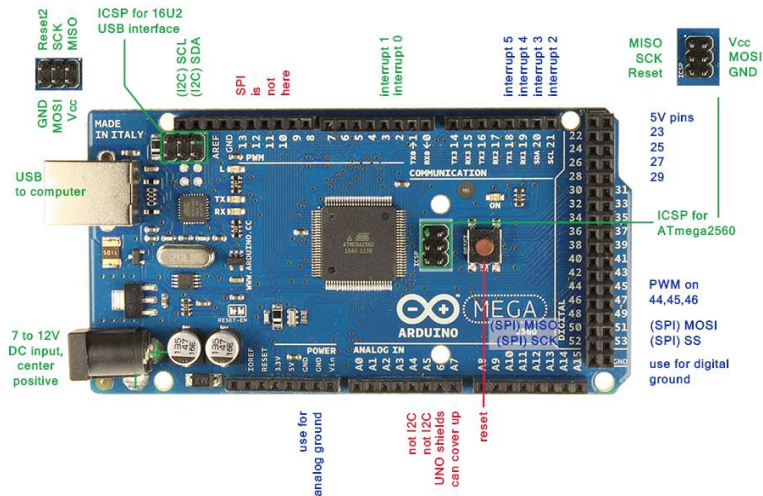
control, mesin kantor, peralatan rumah tangga, alat berat, dan mainan. Dengan mengurangi ukuran, biaya, dan konsumsi tenaga dibandingkan dengan mendesain menggunakan mikroprosesor memori, dan alat *input / output* yang terpisah, kehadiran mikrokontroler membuat kontrol elektrik untuk berbagai proses menjadi lebih ekonomis. Dengan penggunaan *microcontroller* ini maka :

- Sistem elektronik akan menjadi lebih ringkas.
- Rancang bangun sistem elektronik akan lebih cepat karena sebagian besar dari sistem adalah perangkat lunak yang mudah dimodifikasi.
- Pencarian gangguan lebih mudah ditelusuri karena sistemnya yang kompak.

Agar sebuah *microcontroller* dapat berfungsi, maka *microcontroller* tersebut memerlukan komponen eksternal yang kemudian disebut dengan sistem minimum. Untuk membuat sistem minimal paling tidak dibutuhkan sistem *clock* dan reset, walaupun pada beberapa *microcontroller* sudah menyediakan sistem *clock* internal, sehingga tanpa rangkaian eksternal pun *microcontroller* sudah beroperasi.

Arduino Mega 2560 adalah papan *microcontroller* berbasis Atmega 2560. Arduino Mega 2560 54 pin digital *input / output*, dimana 15 pin dapat digunakan sebagai *output* PWM, 16 pin sebagai input analog, dan 4 pin sebagai UART (port serial *hardware*), 16 MHz kristal osilator, koneksi USB, *jack power*, header ICSP, dan tombol reset. Ini semua yang diperlukan untuk mendukung *microcontroller*. Cukup dengan menghubungkannya ke komputer melalui kabel USB atau power dihubungkan dengan adaptor AC – DC atau baterai untuk mulai mengaktifkannya. Arduino Mega 2560 kompatibel dengan sebagian besar shield yang dirancang untuk Arduino Duemilanove atau Arduino Diecimila. Arduino Mega 2560 adalah versi terbaru yang menggantikan versi Arduino Mega.

Arduino Mega 2560 berbeda dari papan sebelumnya, karena versi terbaru sudah tidak menggunakan chip driver FTDI USB-to-serial. Tapi, menggunakan chip Atmega 16U2 (Atmega 8U2 pada papan Revisi 1 dan Revisi 2) yang diprogram sebagai konverter USB-to-serial. Arduino Mega 2560 Revisi 2 memiliki resistor penarik jalur HWB 8U2 ke *Ground*, sehingga lebih mudah untuk dimasukkan ke dalam mode DFU. Berikut merupakan tampilan pinout dari Arduino Mega 2560:



Gambar 2.12 Pinout Arduino Mega 2560

Arduino Mega 2560 Revisi 3 seperti gambar 2.12 memiliki fitur-fitur baru berikut:

- Pinout : Ditambahkan pin SDA dan pin SCL yang dekat dengan pin AREF dan dua pin baru lainnya ditempatkan dekat dengan pin RESET, IOREF memungkinkan *shield* untuk beradaptasi dengan tegangan yang tersedia pada papan. Di masa depan, *shield* akan kompatibel baik dengan papan yang menggunakan AVR yang beroperasi dengan 5 Volt dan dengan Arduino *Due* yang beroperasi dengan tegangan 3.3 Volt. Dan ada dua pin yang tidak terhubung, yang disediakan untuk tujuan masa depan.
- Sirkuit RESET.
- Chip ATmega 16U2 menggantikan chip ATmega 8U2. Dan berikut merupakan spesifikasi dari Arduino Mega 2560 seperti tabel 2.1.

Tabel 2.1 Spesifikasi Dari Arduino Mega 2560

Microcontroller	ATmega 2560
Tegangan Operasi	5V
<i>Input Voltage</i> (disarankan)	7-12V

<i>Input Voltage</i> (limit)	6-20V
Jumlah pin I/O digital	54 (15 pin digunakan sebagai output PWM)
Jumlah pin <i>input</i> analog	16
Arus DC tiap pin I/O	40 mA
Arus DC untuk pin 3.3V	50 mA
<i>Flash Memory</i>	256 KB (8 KB digunakan untuk <i>bootloader</i>)
SRAM	8 KB
EEPROM	4 KB
<i>Clock Speed</i>	16 MHz

Arduino Mega dapat diaktifkan melalui koneksi USB atau dengan catu daya eksternal. Sumber daya dipilih secara otomatis. Sumber daya eksternal (non-USB) dapat berasal baik dari adaptor AC – DC atau baterai. Adaptor dapat dihubungkan dengan mencolokkan steker 2,1 mm yang bagian tengahnya terminal positif ke *jack* sumber tegangan pada papan. Jika tegangan berasal dari baterai dapat langsung dihubungkan melalui *header* pin GND dan pin Vin dari konektor POWER. [2]

Papan Arduino ATmega 2560 dapat beroperasi dengan pasokan daya eksternal 6 Volt sampai 20 Volt. Jika diberi tegangan kurang dari 7 Volt, maka pin 5 Volt mungkin akan menghasilkan tegangan kurang dari 5 Volt dan ini akan membuat papan menjadi tidak stabil. Jika sumber tegangan menggunakan lebih dari 12 Volt, regulator tegangan akan mengalami panas berlebihan dan bisa merusak papan. Rentang sumber tegangan yang dianjurkan adalah 7 Volt sampai 12 Volt. Pin tegangan yang tersedia pada papan Arduino adalah sebagai berikut :

- **VIN** : Adalah *input* tegangan untuk papan Arduino ketika menggunakan sumber daya eksternal (sebagai “saingan” tegangan 5 Volt dari koneksi USB atau sumber daya ter – regulator lainnya). Anda dapat memberikan tegangan melalui pin ini, atau jika memasok tegangan untuk papan melalui *jack power*, kita bisa mengakses / mengambil tegangan melalui pin ini.
- **5V** : Sebuah pin yang mengeluarkan tegangan ter – regulator 5 Volt, dari pin ini tegangan sudah diatur (ter – regulator) dari regulator yang tersedia (built-in) pada papan. Arduino dapat diaktifkan dengan sumber daya baik berasal dari *jack power* DC (7-12 Volt), konektor USB (5 Volt), atau pin Vin pada board (7-

12 Volt). Memberikan tegangan melalui pin 5V atau 3,3V secara langsung tanpa melewati regulator dapat merusak papan Arduino.

- **3V3** : Sebuah pin yang menghasilkan tegangan 3,3 Volt. Tegangan ini dihasilkan oleh regulator yang terdapat pada papan (on-board). Arus maksimum yang dihasilkan adalah 50 mA.
- **GND** : Pin *Ground* atau Massa.
- **IOREF** : Pin ini pada papan Arduino berfungsi untuk memberikan referensi tegangan yang beroperasi pada *microcontroller*. Sebuah perisai (*shield*) dikonfigurasi dengan benar untuk dapat membaca pin tegangan IOREF dan memilih sumber daya yang tepat atau mengaktifkan penerjemah tegangan (*voltage translator*) pada output untuk bekerja pada tegangan 5 Volt atau 3,3 Volt.

Untuk memori, Arduino ATmega 2560 memiliki 256 KB *flash memory* untuk menyimpan kode (8 KB digunakan untuk *bootloader*), 8 KB SRAM dan 4 KB EEPROM (yang dapat dibaca dan ditulis dengan perpustakaan EEPROM).

Sedangkan *input* dan *output* dari Arduino Atmega 2560, masing – masing dari 54 digital pin pada Arduino Mega dapat digunakan sebagai *input* atau *output*, menggunakan fungsi *pinMode()*, *digitalWrite()*, dan *digitalRead()*. Arduino Mega beroperasi pada tegangan 5 volt. Setiap pin dapat memberikan atau menerima arus maksimum 40 mA dan memiliki resistor pull-up internal (yang terputus secara *default*) sebesar 20 – 50 kΩ. Selain itu, beberapa pin memiliki fungsi khusus, antara lain :

- Serial yang digunakan untuk menerima (RX) dan mengirimkan (TX) data serial TTL seperti terlihat pada tabel 2.2 dan tabel 2.3.

Tabel 2.2 Tabel Pin Serial RX dan TX

Nomor Pin	Nama Pin	Peta Nama Pin
2	PE0 (RXD0/PCINT8)	Digital pin 0 (RX0)
3	PE1 (TXD0)	Digital pin 1 (TX0)
12	PH0 (RXD2)	Digital pin 17 (RX2)
13	PH1 (TXD2)	Digital pin 16 (TX2)
45	PD2 (RXDI/INT2)	Digital pin 19

		(RX1)
46	PD3 (TXD1/INT3)	Digital pin 18 (TX1)
63	PJ0 (RXD3/PCINT9)	Digital pin 15 (RX3)
64	PJ1 (TXD3/PCINT10)	Digital pin 14 (TX3)

- Eksternal Interupsi : Pin ini dapat dikonfigurasi untuk memicu sebuah interupsi pada nilai yang rendah, meningkat atau menurun, atau berubah nilai.

Tabel 2.3 Tabel Pin Eksternal Interupsi

Nomor Pin	Nama Pin	Peta Nama Pin
6	PE4 (OC3B/INT4)	Digital pin 2 (PWM)
7	PE5 (OC3C/INT5)	Digital pin 3 (PWM)
43	PD0 (SCL/INT0)	Digital pin 21 (SCL)
44	PD1 (SDA/INT1)	Digital pin 20 (SDA)
45	PD2 (RXDI/INT2)	Digital pin 19 (RX1)
46	PD3 (TXD1/INT3)	Digital pin 18 (TX1)

- **SPI** : Pin ini mendukung komunikasi SPI menggunakan SPI library. Pin SPI juga terhubung dengan header ICSP, yang secara fisik kompatibel dengan Arduino *Uno*, Arduino *Duemilanove* dan Arduino *Diecimila* seperti terlihat pada tabel 2.4.

Tabel 2.4 Tabel Pin *SPI*

Nomor Pin	Nama Pin	Peta Nama Pin
19	PB0 (SS/PCINT0)	Digital pin 53 (SS)
20	PB1 (SCK/PCINT1)	Digital pin 52 (SCK)

21	PB2 (MOSI/PCINT2)	Digital pin 51 (MOSI)
22	PB3 (MISO/PCINT3)	Digital pin 50 (MISO)

- **LED** : Pin 13. Tersedia secara built-in pada papan Arduino ATmega LED terhubung ke pin digital 13. Ketika pin diset bernilai HIGH, maka LED menyala (ON), dan ketika pin diset bernilai LOW, maka LED padam (OFF).
- **TWI** : Pin 20 (SDA) dan pin 21 (SCL). Yang mendukung komunikasi TWI menggunakan *Wire Library*. Perhatikan bahwa pin ini tidak di lokasi yang sama dengan pin TWI pada Arduino *Duemilanove* atau Arduino *Diecimila*.

Arduino Mega 2560 memiliki 16 pin sebagai analog *input*, yang masing – masing menyediakan resolusi 10 bit (yaitu 1024 nilai yang berbeda). Secara default pin ini dapat diukur / diatur dari mulai *Ground* sampai dengan 5 Volt, juga memungkinkan untuk mengubah titik jangkauan tertinggi atau terendah mereka menggunakan pin AREF dan fungsi *Analog Reference()*. Ada beberapa pin lainnya yang tersedia, antara lain :

- AREF : Referensi tegangan untuk *input* digunakan dengan fungsi *Analog Reference()*.
- RESET: Jalur LOW ini digunakan untuk me – reset (menghidupkan ulang) *microcontroller*. Jalur ini biasanya digunakan untuk menambahkan tombol reset pada *shield* yang menghalangi papan utama Arduino.

Arduino Mega 2560 memiliki sejumlah fasilitas untuk berkomunikasi dengan komputer, dengan Arduino lain, atau dengan *microcontroller* lainnya. Arduino ATmega 328 menyediakan 4 hardware komunikasi serial UART TTL (5 Volt). Sebuah chip ATmega 16U2 (ATmega 8U2 pada papan Revisi 1 dan Revisi 2) yang terdapat pada papan digunakan sebagai media komunikasi serial melalui USB dan muncul sebagai COM Port Virtual (pada Device komputer) untuk berkomunikasi dengan perangkat lunak pada komputer, untuk sistem operasi Windows masih tetap memerlukan file inf, tetapi untuk sistem operasi OS X dan Linux akan mengenali papan sebagai port COM secara otomatis. Perangkat lunak Arduino termasuk didalamnya serial monitor memungkinkan data tekstual sederhana dikirim ke dan dari papan Arduino. LED RX dan TX yang tersedia pada papan akan

berkedip ketika data sedang dikirim atau diterima melalui chip USB-to-serial yang terhubung melalui USB komputer (tetapi tidak untuk komunikasi serial seperti pada pin 0 dan 1).

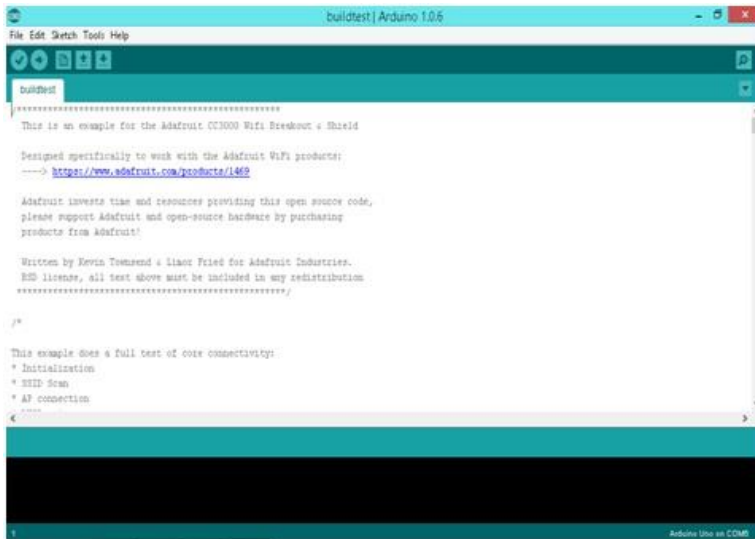
Sebuah *Serial Library Software* memungkinkan untuk komunikasi serial pada salah satu pin digital Mega 2560. ATmega 2560 juga mendukung komunikasi TWI dan SPI. Perangkat lunak Arduino termasuk *Wire Library* digunakan untuk menyederhanakan penggunaan bus TWI. Untuk komunikasi SPI, menggunakan *SPI library*.

IDE (*Integrated Development Environment*) Arduino merupakan aplikasi yang mencakup editor, *compiler*, dan *uploader* dapat menggunakan semua seri modul keluarga Arduino, seperti Arduino *Duemilanove*, *Uno*, *Bluetooth*, *Mega*. Kecuali ada beberapa tipe board produksi Arduino yang memakai *microcontroller* di luar seri AVR, seperti mikroprosesor ARM. Saat menulis kode program atau mengkompilasi modul *hardware* Arduino tidak harus tersambung ke PC atau *Notebook*, walaupun saat proses unggahan ke board diperlukan modul *hardware*.

IDE Arduino juga memiliki keterbatasan tidak mendukung fungsi *debugging hardware* maupun *software*. Proses kompilasi IDE Arduino diawali dengan proses pengecekan kesalahan sintaksis *sketch*, kemudian memanfaatkan pustaka *Processing* dan *avr - gcc sketch* dikompilasi menjadi berkas object, lalu berkas - berkas object digabungkan oleh pustaka Arduino menjadi berkas biner. Berkas biner ini diunggah ke chip *microcontroller* via kabel USB, serial port DB9, atau Serial Bluetooth.

Compiler IDE Arduino juga memanfaatkan pustaka *open source* AVRLibc sebagai standar *de-facto* pustaka referensi dan fungsi register *microcontroller* AVR. Pustaka AVRLibc ini sudah disertakan dalam satu paket program IDE Arduino. Meskipun demikian, kita tidak perlu mendefinisikan *directive#include* dari pustaka AVRLibc pada *sketch* karena otomatis compiler me-link pustaka AVRLibc tersebut.

Ukuran berkas biner HEX hasil kompilasi akan semakin besar jika kode *sketch* semakin kompleks. Berkas biner memiliki ekstensi *.hex* berisi data instruksi program yang biasa dipahami oleh *microcontroller* target. Selain itu, port paralel juga bisa dipakai untuk mengunggah *bootloader* ke *microcontroller*. Meskipun demikian, cara ini sudah jarang digunakan karena sekarang hampir tidak ada *mainboard* PC yang masih menyediakan port paralel, dan pada *notebook* juga sudah tidak menyertakan port paralel.



Gambar 2.13 Tampilan *Sketch* di Arduino IDE

Pada Gambar 2.13 Terlihat *button* (tombol) yang ada di IDE Arduino, *button compile* berfungsi untuk kompilasi *sketch* tanpa unggah ke board bisa dipakai untuk pengecekan kesalahan kode sintaks *sketch*. *Button upload* untuk mengunggah hasil kompilasi *sketch* ke board target. Pesan *error* akan terlihat jika board belum terpasang atau alamat port COM belum terkonfigurasi dengan benar.

Berkas Pustaka yang tersimpan di dalam direktori yang sama *sketchbook* akan terlihat dalam *Tab sketchbook*. Berkas pustaka yang tersimpan di direktori / Arduino / *libraries* / tidak ditampilkan pada tab *sketch* meskipun bias diakses oleh *sketch* lain.

Daripada menekan tombol reset sebelum upload, Arduino Mega 2560 didesain dengan cara yang memungkinkan Anda untuk me – reset melalui perangkat lunak yang berjalan pada komputer yang terhubung. Salah satu jalur kontrol hardware (DTR) mengalir dari ATmega 8U2 / 16U2 dan terhubung ke jalur reset dari ATmega 2560 melalui kapasitor 100 *nanofarad*. Bila jalur ini di-set rendah / *low*, jalur *resetdrop* cukup lama untuk me-reset chip. Perangkat lunak Arduino menggunakan kemampuan ini untuk memungkinkan Anda meng-upload kode dengan hanya menekan tombol upload pada perangkat lunak Arduino. Ini berarti

bahwa *bootloader* memiliki rentang waktu yang lebih pendek, seperti menurunkan DTR dapat terkoordinasi (berjalan beriringan) dengan dimulainya upload.

Pengaturan ini juga memiliki implikasi lain. Ketika Arduino Mega 2560 terhubung dengan komputer yang menggunakan sistem operasi Mac OS X atau Linux, papan Arduino akan di-reset setiap kali dihubungkan dengan *software* komputer (melalui USB). Dan setengah detik kemudian atau lebih, *bootloader* berjalan pada papan Mega 2560. Proses reset melalui program ini digunakan untuk mengabaikan data yang cacat (yaitu apapun selain meng-upload kode baru), ia akan memotong dan membuang beberapa *byte* pertama dari data yang dikirim ke papan setelah sambungan dibuka. Jika sebuah sketsa dijalankan pada papan untuk menerima satu kali konfigurasi atau menerima data lain ketika pertama kali dijalankan, pastikan bahwa perangkat lunak diberikan waktu untuk berkomunikasi dengan menunggu satu detik setelah terkoneksi dan sebelum mengirim data.

Mega 2560 memiliki trek jalur yang dapat dipotong untuk menonaktifkan fungsi *auto – reset*. Pad di kedua sisi jalur dapat hubungan dengan disolder untuk mengaktifkan kembali fungsi *auto – reset* Pad berlabel “RESET-EN”. Anda juga dapat menonaktifkan *auto – reset* dengan menghubungkan resistor 110 ohm dari 5V ke jalur reset.

Selain itu Arduino Mega 2560 memiliki *polyfuse* reset yang melindungi port USB komputer anda dari hubungan singkat dan arus lebih. Meskipun pada dasarnya komputer telah memiliki perlindungan internal pada port USB mereka sendiri, sekring memberikan lapisan perlindungan tambahan. Jika arus lebih dari 500 mA dihubungkan ke port USB, sekring secara otomatis akan memutuskan sambungan sampai hubungan singkat atau *overload* dihapus / dibuang.

2.6 Input Output Analog

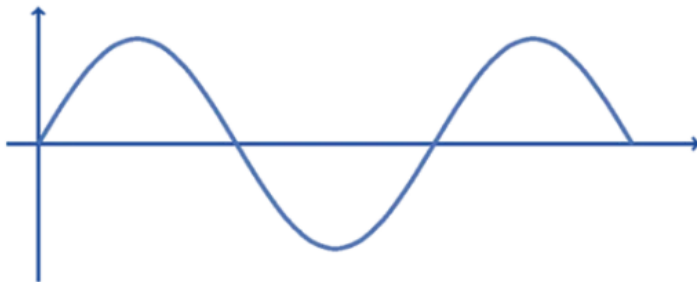
Pada Arduino mega terdapat pin untuk mengakses atau menerima data yang berupa *input output* analog dan *input output* digital. Penjelasan untuk keduanya yaitu:[9]

A. Input Anaog

Sinyal analog menjelaskan variabel fisik yang bervariasi terus menerus yang berhubungan dengan variabel lain seperti gambar 2.14. Contoh dari sinyal analog yaitu intensitas cahaya LED yang dimana arus yang didalamnya juga meningkat, arus melalui resistor seperti yang kita memvariasikan tegangan, atau

suhu di kamar yang berjalan sering dengan berjalannya waktu. Sinyal analog dapat mengambil nilai-nilai yang tak terbatas. Sayangnya mikrokontroler tidak dapat mewakili jumlah yang tak terbatas nilainya itu. Jadi yang harus dilakukan ketika ada sinyal analog dengan cara sampling dengan cara mengubah sinyal analog tersebut menjadi sinyal digital melalui pin khusus yaitu pin Analog ke Digital Konversi (ADC). Ini pada dasarnya berarti bahwa memilih nilai-nilai tertentu di suatu daerah dan kemudian kita menggunakan serangkaian bit untuk mewakili nilai-nilai tersebut saja.

Pin analog (ADC) tersebut berfungsi sebagai mengubah sinyal analog yang masuk menjadi nilai digital sehingga mudah diolah dan dapat diukur. Pin analog dapat mengenali sinyal pada rentang nilai voltase tersebut (normal 0 – 5 Vdc) yang dibandingkan dengan nilai tegangan referensinya. Hal ini sangat berguna ketika kita hendak mengukur nilai dari suatu sensor yang terpasang serta menggunakan nilai masukan tersebut untuk keperluan lain. Fungsi yang kita gunakan untuk membaca nilai analog pada Arduino adalah `analogRead([nomorPin])`. [9]

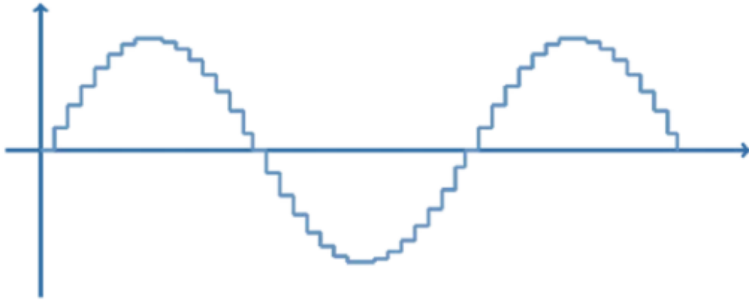


Gambar 2.14 Tampilan Sinyal *Input* Analog

B. *Output* Analog

Pada *output* analog Arduino tidak dapat dihasilkan secara langsung, tetapi harus melewati proses pengubahan *output* dari digital menjadi analog yang memerlukan fungsi komponen *Digital to Analog Converter*. Sayangnya pada Arduino fungsi tersebut tidak ada, sehingga memerlukan modul eksternal

(modul DAC) sebagai konverter sinyalnya, contoh sinyal *output* analog dapat di lihat pada gambar 2.15. [9]



Gambar 2.15 Tampilan Sinyal *Output* Analog

Tetapi disisi lain, output analog pada Arduino kebanyakan memiliki fitur PWM (*Pulse Width Modulation*). Contoh kasus apabila arduino yang dihubungkan dengan LED ingin mengendalikan intensitas cahaya LED fitur ini dapat digunakan. Analog *output* pada Arduino mengirimkan sinyal analog dengan intensitas yang ditentukan sesuai kebutuhan. PWM memanipulasi keluaran digital sedemikian rupa sehingga menghasilkan sinyal analog. Arduino mengeset *output* digital ke *HIGH* dan *LOW* bergantian dengan rentang waktu tertentu untuk setiap nilai keluarannya. Durasi waktu untuk nilai *HIGH* disebut *pulse width* atau panjang pulsa. Variasi nilai *output* analog didapatkan dari perubahan panjang pulsa yang diberikan pada satu periode waktu dan dilakukan berulang-ulang seperti gambar 2.16.

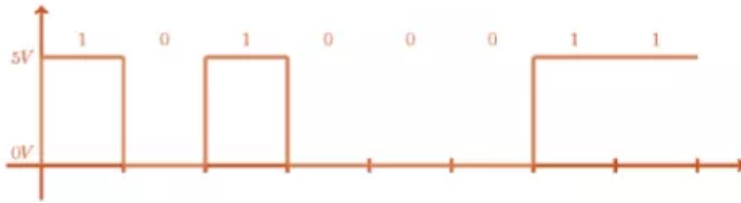


Gambar 2.16 Tampilan PWM Setelah Diberikan Filter

C. Input Digital

Pin pada Arduino, baik pin digital maupun pin analog, dapat digunakan sebagai pin I/O (*input Output*) digital. Digital berarti sinyal yang akan dikirimkan/diterima berupa nilai 1 atau 0, kondisi *on* atau *off*, kondisi *HIGH* atau *LOW*, ataupun ada atau tidak ada sinyal. Berbeda dengan sinyal analog yang nilainya

bersifat kontinyu, yakni nilai antara 0 dan 1 bernilai dipertimbangkan. Pin digital berarti pin dapat menerima/mengirim sinyal digital. Contoh dari input digital yaitu penggunaan sensor PIR dimana output dari sensor tersebut berupa nilai 1 jika terdeteksi adanya benda bergerak dan nilai 0 jika tidak terdeteksi adanya benda seperti gambar2.17.



Gambar 2.17 Tampilan Sinyal *Input* Digital

D. Output Digital

Pada *output* digital sebenarnya hampir sama dengan *input* digital dikarenakan pada dasarnya sama, hal yang dikirimkan yaitu nilai 1 atau nilai 0. Yang membedakan hanya pada penggunaan fungsi pada saat pemogramannya, kapan saat diset sebagai *input* maupun *output*.

format dasar pemograman untuk deklarasi >> pinMode ([pin yang digunakan], [INPUT or OUTPUT]);

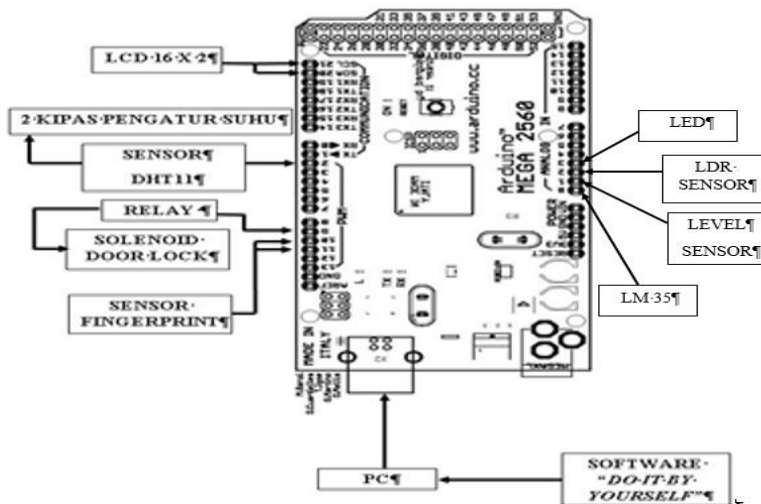
BAB III

PERENCANAAN DAN PEMBUATAN ALAT

Pada bab ini berisi tahapan mengenai tahapan yang dilakukan dalam perencanaan dan pembuatan Tugas Akhir. Penjelasan diawali dengan blok fungsional sistem secara keseluruhan yang meliputi proses kerja alat dalam bentuk alur diagram. Perancangan mekanik yang membahas tentang desain dan pembuatan mekanik yang mendukung cara kerja alat. Perancangan elektrik yang membahas perancangan rangkaian elektrik sebagai rangkaian pendukung alat. Serta perancangan perangkat lunak yang meliputi perancangan diagram alur program dan desain *software* “*Do It by Yourself*” menggunakan Microsoft Visual Studio.

3.1 Blok Fungsional Sistem

Sebelum melakukan perancangan perangkat keras dan perangkat lunak, diperlukan sebuah perancangan blok fungsional sistem berupa blok diagram yang menjelaskan sistem kerja secara keseluruhan Tugas Akhir ini. Secara keseluruhan blok fungsional sistem dapat dilihat pada gambar 3.1 berikut.



Gambar 3.1 Pinout Arduino Mega 2560

Dari gambar 3.1 tersebut, maka dapat dijelaskan tentang pemetaan pinout atau I/O dari Arduino yang menghubungkan ke masing – masing komponen dari alat maupun sensor serta *software interface* yang digunakan oleh *user* atau orang awam yang menginginkan nilai atau fungsi dari I/O Arduino tanpa harus memprogram terlebih dahulu yang tentunya membutuhkan pendalaman pada bahasa pemrograman Arduino tersebut.

Dimana output yang pertama dari output Arduino yaitu sensor *fingerprint*. Cara kerja dari sensor *fingerprint* ini yaitu dengan “merekam” atau mengenali dengan cara membaca pola sidik jari dari *user* yang kemudian akan disimpan oleh Arduino. Lalu selanjutnya proses identifikasi tersebut bisa dilakukan dengan memverifikasi atau mencocokkan data atau pola sidik jari dari *user* tersebut yang sebelumnya telah discan untuk disimpan, dan jika data dinyatakan sama, maka akses akan otomatis terbuka. Kemudian dari proses pengenalan oleh *fingerprint*, maka dilanjutkan ke sensor solenoid *door lock*. Yang dimana sensor ini memiliki cara kerja dimana saat solenoid NC (*Normally Close*) diberi tegangan, maka solenoid akan memanjang atau tertutup. Dan untuk solenoid NO (*Normally Open*) memiliki cara kerja kebalikan dari solenoid NC. Begitu juga dari *output* Arduino atau menerima data dari sensor *fingerprint*. Jika data pengenalan *user* sama dengan inputan dari memori sebelumnya atau pengenalan yang dilakukan sebelumnya, maka solenoid akan memendek sehingga *prototype* bisa dibuka, begitu juga dengan kebalikannya jika data atau *user* berbeda maka akses akan ditolak.

Output dari Arduino selanjutnya yaitu digunakan untuk sensor suhu dan kelembaban atau sensor DHT11. Dimana cara kerja atau fungsi dari sensor DHT11 yaitu mengontrol dan menjaga kestabilan suhu dan kelembaban yang dibutuhkan serta mengatur sirkulasi dan kebersihan udara di dalam *prototype* rancangan alat Tugas Akhir ini. Karena masing – masing komponen maupun isi di dalam *prototype* alat perlu dijaga suhu dan kelembabannya karena rata – rata material atau bahan dari komponen terbuat dari logam yang tentunya memiliki sifat mudah korosif atau berkarat. Setelah data nilai atau variabel dari suhu dan kelembaban yang dibutuhkan dari dalam “ruang” *prototype* tersebut disimpan yang memiliki suhu dan kelembaban minimal maupun maksimal. Jika pada saat nilai suhu di dalam *prototype* tersebut melebihi maupun mendekati nilai maksimum atau pada saat suhu di dalam *prototype* tersebut dingin, maka secara otomatis akan menyalakan dua

kipas yang ditempatkan diantara dalam ruang dari *prototype* serta luar ruang dari *prototype*, sehingga bisa menjaga sirkulasi udara demi meminimalisir proses korosif atau berkarat karena adanya nilai kelembaban yang tinggi.

Selanjutnya, dari 3 data diatas akan ditampilkan di tampilan LCD 16x2 yang ditempatkan di bagian depan dari *prototype* rancangan alat Tugas Akhir ini. Dimana untuk tampilan dari sensor *fingerprnt* yang menampilkan data dari *user* yang sebelumnya telah disimpan atau dikenali dan akses untuk bisa membuka dan akses *error* atau ketika akses ditolak karena data tidak sesuai dengan pengenalan dari *user*.

Selain itu juga LCD 16x2 diprogram untuk menampilkan data dari sensor DHT11 yang memonitoring suhu maupun kelembaban didalam *prototype* rancangan Tugas Akhir.

Dari keseluruhan output Arduino yang telah dijelaskan diatas yang mencakup cara kerja maupun fungsi dari masing – masing komponen, tentunya dibutuhkan program dari Arduino atau dalam bahasa pemrograman Arduino biasa disebut dengan *library*, dimana *library* tersebut disesuaikan dengan *output* dari Arduino yang berupa sensor maupun komponen – komponen tertentu.

Dalam *prototype* rancangan Tugas Akhir ini, telah dibuat *database* program dalam bahasa pemrograman Arduino atau dengan kata lain seperti yang dijelaskan tadi yaitu *library*. Tentunya untuk melakukan pemrograman ke dalam Arduino membutuhkan pendalaman terlebih dahulu terhadap bahasa pemrograman Arduino, dan juga tentunya tidak semua orang bisa melakukan pemrograman dari *hardware* Arduino ini.

Maka dari itu, dalam rancangan Tugas Akhir ini dibuatlah suatu *software* dengan menggunakan bahasa pemrograman Visual Basic yang dimana dirancang untuk mempermudah penggunaan rancangan Tugas Akhir ini kepada para pengguna yang rata – rata orang awam yang belum tahu seluk beluk dari bahasa pemrograman Arduino yang ingin menentukan sendiri secara bebas atau sesuai keinginan fungsi dari *I/O* dari Arduino yang telah dirancang dalam rancangan *prototype* alat Tugas Akhir ini atau biasa disebut dengan “*Do It by Yourself*”.

3.2 Kelebihan dan Kekurangan Komponen Pada Software

Pada sub bab ini akan dibahas mengenai Kelebihan dan Kekurangan pada spesifikasi masing – masing komponen yang dipakai untuk *software* ‘*Do It By Yourself*’ rancangan alat Tugas Akhir ini.

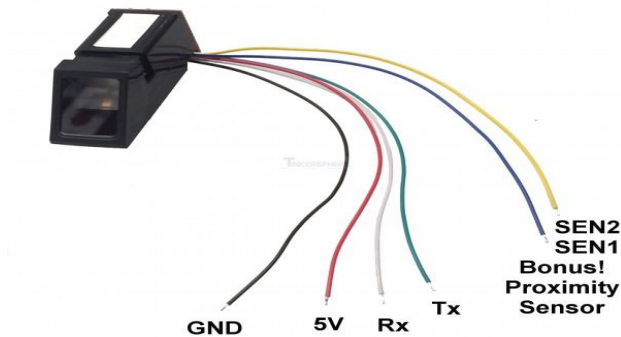
- Sensor *Fingerprint Optical R305*

Sensor *fingerprint* memiliki jenis – jenis tersendiri yang mempunyai spesifikasi masing – masing dan tentunya memiliki kelebihan dan kekurangan tersendiri. Didalam *prototype* rancangan alat Tugas Akhir ini memakai sensor *fingerprint optical R305* seperti gambar 3.2 yang memiliki kelebihan sebagai berikut.

1. Pembaca sidik jari biometrik optik dengan fitur lebih canggih dan bisa disematkan ke dalam berbagai produk akhir, seperti: kontrol akses, kehadiran, brankas, kunci pintu mobil.
2. Pengambilan gambar yang tidak terintegrasi dan chip algoritma bersama-sama, *ALL-in-One*.
3. Pembaca sidik jari bisa melakukan pengembangan sekunder, bisa disematkan ke dalam berbagai produk akhir
4. Konsumsi daya rendah, biaya rendah, ukuran kecil, performa prima.
5. Teknologi optik profesional, teknik pembuatan modul yang tepat.
6. Kemampuan pengolahan gambar yang bagus, bisa berhasil menangkap gambar hingga resolusi 500 DPI (Dots per Inch).

Dan berikut merupakan kelebihan sensor *fingerprint optical SM630* yang memiliki spesifikasi maupun fisik yang hampir sama dengan sensor *fingerprint R305*.

1. Memiliki intelektual tersendiri (kemampuan yang dibutuhkan untuk melakukan berbagai aktivitas yang dibutuhkan pada aplikasi alat seperti membaca/merekam pola sidik jari, menyimpan data, mengupload, dll).
2. Berbagai macam aplikasi sidik jari dengan kualitas berbeda.
3. Algoritma peningkatan yang sangat besar.
4. Penggunaannya lebih fleksibel.
5. Mudah digunakan dan dapat dikembangkan pengaplikasiannya.
6. Konsumsi daya rendah.



Gambar 3.2 Sensor Fingerprint Optical SM630

Dari perbandingan spesifikasi dan kelebihan dari kedua sensor fingerprint tersebut, dapat diambil kesimpulan yaitu sama – sama berjenis sensor *fingerprint optical* dan memiliki konsumsi daya rendah serta pengimplementasian dari kedua sensor tersebut dapat dikembangkan sesuai dengan kebutuhan dari pengguna sensor tersebut, hanya saja sensor fingerprint R305 memiliki harga yang lebih terjangkau dibandingkan dengan sensor *fingerprint* SM630.

- **Sensor LM 35**

Dalam *software* disediakan pilihan sensor yang bisa mengontrol maupun memonitoring suhu dan dipilih lah sensor LM 35, tentu pemilihan sensor ini mempunyai kelebihan dan kekurangan diantaranya.

- **Kelebihan:**

- a. Rentang suhu yang jauh, antara -55 sampai +150 oC.
- b. Low self-heating, sebesar 0.08 oC.
- c. Beroperasi pada tegangan 4 sampai 30 V.
- d. Rangkaian tidak rumit.
- e. Tidak memerlukan pengkondisian sinyal.

- **Kekurangan:**

Membutuhkan sumber tegangan untuk beroperasi.

LM35 adalah komponen sensor suhu berukuran kecil seperti transistor (TO-92). Komponen yang sangat mudah

digunakan ini mampu mengukur suhu hingga 100 derajat Celcius, tetapi tidak cocok untuk pengukur suhu yang sensornya dimasukan dalam cairan. Dengan tegangan keluaran yang terskala linear dengan suhu terukur, yakni 10 milivolt per 1 derajat Celcius, maka komponen ini sangat cocok untuk digunakan sebagai eksperimen kita, atau bahkan untuk aplikasi-aplikasi seperti termometer ruang digital, mesin pasteurisasi, atau termometer badan digital. LM35DZ adalah komponen sensor suhu berukuran kecil seperti transistor (TO-92). Komponen yang sangat mudah digunakan ini mampu mengukur suhu hingga 100 derajat Celcius. Dengan tegangan keluaran yang terskala linear dengan suhu terukur, yakni 10 milivolt per 1 derajat Celcius, maka komponen ini sangat cocok untuk digunakan sebagai teman eksperimen kita, atau bahkan untuk aplikasi-aplikasi seperti termometer ruang digital, mesin pasteurisasi, atau termometer badan digital.

LM35 dapat disuplai dengan tegangan mulai 4V-30V DC dengan arus pengurasan 60 mikroampere. IC LM 35 sebagai sensor suhu yang teliti dan terkemas dalam bentuk Integrated Circuit (IC), dimana output tegangan keluaran sangat linear terhadap perubahan suhu. Sensor ini berfungsi sebagai penguubah dari besaran fisis suhu ke besaran tegangan yang memiliki koefisien sebesar 10 mV /°C yang berarti bahwa kenaikan suhu 1° C maka akan terjadi kenaikan tegangan sebesar 10 mV. Seperti sensor suhu jenis RTD PT100 atau PT1000 misalnya, komponen ini tidak boleh dieksitasi oleh arus melebihi 1 miliampere, jika melebihi, maka sensor akan mengalami self-heating yang menyebabkan hasil pengukuran senantiasa lebih tinggi dibandingkan suhu yang sebenarnya.

- **Sensor Pengukuran Level**

Satuan level umumnya meter (m). Namun, ada banyak cara untuk mengukur level menggunakan teknologi yang berbeda dan unit pengukuran yang bervariasi.

Diantaranya adalah:

- Ultrasonic, waktu transit.
- Pulse echo.
- Pulse radar.

- Tekanan, hidrostatik.
- Berat, tegangan gauge.
- Konduktivitas.
- Kapaistif.

Untuk pengukuran kontinu, level terdeteksi dan diubah menjadi sinyal yang sebanding dengan tingkatannya. Mikroprosesor *based devices* dapat menunjukkan panjang atau volume.

Teknik yang berbeda juga memiliki kebutuhan yang berbeda. Misalnya, ketika mendeteksi level dari bagian atas tanki, bentuk tangki diperhitungkan untuk mencari volumenya.

Bila menggunakan cara hidrostatik, yang mendeteksi tekanan dari bagian bawah.

tangki, maka densitas harus diketahui dan harus konstan.

Keuntungan

- Sangat sederhana.
- Murah.

Kekurangan.

- Tidak cocok untuk kontrol otomatis.
- Memerlukan pembersihan.
- Mudah rusak.




- **Sensor LDR**

Sensor Cahaya LDR (Light Dependent Resistor) adalah salah satu jenis resistor yang dapat mengalami perubahan resistansinya apabila mengalami perubahan penerimaan cahaya. Besarnya nilai hambatan pada Sensor Cahaya LDR (Light Dependent Resistor) tergantung pada besar kecilnya cahaya yang diterima oleh LDR itu sendiri. LDR sering disebut dengan alat atau sensor yang berupa resistor yang peka terhadap cahaya. Biasanya LDR terbuat dari cadmium sulfida yaitu merupakan bahan semi konduktor yang resistansinya berubah-ubah menurut banyaknya cahaya (sinar) yang mengenainya. Resistansi LDR pada tempat yang gelap biasanya mencapai sekitar $10\text{M}\Omega$ dan di tempat terang LDR mempunyai resistansi yang turun menjadi sekitar 150Ω . Seperti halnya resistor konvensional, pemasangan LDR dalam suatu rangkaian sama persis seperti pemasangan resistor biasa. LDR memiliki kelebihan dan kekurangan sebagai komponen sensor cahaya.

- Kelebihan dari LDR adalah ia benar-benar hanya tergantung pada seberapa banyak cahaya yang

diterima.hal tersebut mengakibatkan gaya atau gangguan eksternal tidak akan mempengaruhi devais yang menggunakan LDR.LDR juga merupakan sensor cahaya yang konsep penggunaannya paling simpel. Selain itu material penyusunnya merupakan material yang sudah populer dan dikenal luas di berbagai aplikasi teknologi sehingga biaya produksinya menjadi rendah dan tersedia hampir di seluruh daerah.

- Kekurangan yaitu ia tidak dapat mendeteksi cahaya dengan tingkat intensitas yang rendah sehingga mungkin tidak dapat bekerja pada beberapa kondisi. kekurangan lainnya adalah LDR termasuk sensor cahaya yang bersifat lambat dalam mendeteksi cahaya pada kondisi peralihan tingkat intensitas cahaya yang berbeda membutuhkan hingga beberapa detik respon waktu untuk mengenali perbedaan intensitas. hal tersebut dikarenakan elektron pada LDR membutuhkan waktu untuk bergerak saat berpindah baik gerakan mempercepat ataupun memperlambat.
- Sensor DHT11
Dalam *software* disediakan pilihan sensor yang bisa mengontrol maupun memonitoring suhu dan kelembaban didalam “ruang”. Dan untuk sensor suhu dan kelembaban sendiri memiliki berbagai macam dan perbedaan dari fisik maupun spesifikasinya, tetapi dari cara kerjanya, semua sensor ini memiliki beberapa persamaan. Contoh dari sensor suhu dan kelembaban yaitu seperti DHT11, DHT22, serta sensor suhu dan kelembaban dari *type* SHT seperti SHT10, SHT11, SHT15, SHT21, SHT25, SHT71, dan SHT75 seperti gambar 3.3.

Humidity Sensor		Packaging	Max. RH tolerance*	Max. T tolerance*	Sensor Output
SHT10		SMD	±4.5%RH	±0.5°C	Digital Sbus
SHT11		SMD	±3%RH	±0.4°C	Digital Sbus
SHT15		SMD	±2%RH	±0.3°C	Digital Sbus
SHT21		DFN	±3%RH	±0.4°C	I ² C, PWM, SDM
SHT25		DFN	±2%RH	±0.3°C	I ² C
SHT71		Pins	±3%RH	±0.4°C	Digital Sbus
SHT75		Pins	±1.8%RH	±0.3°C	Digital Sbus

Gambar 3.3 Spesifikasi Sensor Suhu dan Kelembaban SHT

DHT11 memiliki *output* digital yang sudah terkalibrasi. Sensor ini terdiri dari komponen pengukur kelembaban tipe *resistive* dan pengukuran suhu via NTC serta terhubung dengan 8 bit uC sehingga memberikan hasil yang cukup baik, kecepatan respon yang cukup, memiliki ketahanan yang baik terhadap interferensi dan cukup murah dalam harga. *Interface* yang digunakan adalah *single write, serial interface* yang cukup cepat dan mudah. Ukuran sensor yang kecil, kebutuhan konsumsi daya yang rendah dan mampu mentransmisikan outputnya dalam jarak 20 meter.

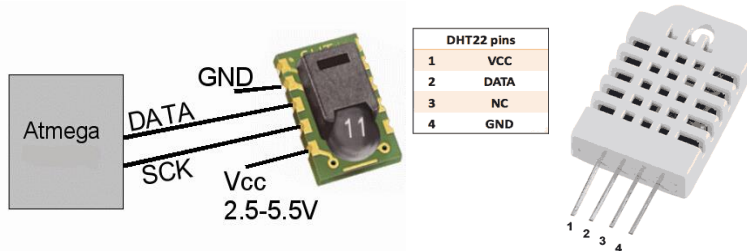
Kelemahan dari sensor DHT11 ini adalah akurasi yang kurang, selain itu range pengukuran suhunya hanya 0 sd 50 derajat celcius tapi dengan harga yang cukup murah sensor ini bisa menjadi alternatif untuk dipakai dalam pengukuran suhu dan kelembaban yang tidak terlalu memerlukan akurasi yang bagus. Misalnya pengukuran suhu kamar, pengukuran suhu dan kelembaban mesin penetas dan lain-lain.

Catu daya yang diperlukan DHT11 ini berkisar 3.5V sampai 5V. Akses ke sensor hanya diperbolehkan lebih dari 1 detik setelah catu daya pertama kali diberikan. Perlu pula ditambahkan kapasitor 100nF diantara pin VCC dan GND untuk filter catu daya.

Sedangkan untuk sensor DHT22 memiliki range pengukuran yang luas yaitu 0 sampai 100% untuk kelembaban

dan -40 derajat celcius sampai 125 derajat celcius untuk suhu. Sensor ini juga memiliki output digital (*single-bus*) dengan akurasi yang tinggi.

Lalu untuk sensor SHT11 memiliki spesifikasi yang lebih baik daripada kedua sensor diatas. Sensor ini dapat mengukur suhu dari -40C hingga +123,8C, atau dari -40F hingga +254,9F dan kelembaban relatif dari 0%RH hingga 1%RH dan memiliki ketetapan (akurasi) pengukuran suhu hingga 0,5C pada suhu 25C dan ketepatan (akurasi) pengukuran kelembaban relatif hingga 3,5%RH, serta membutuhkan catu daya +5V DC dengan konsumsi daya rendah 30 μ W. Modul ini memiliki faktor bentuk 8 pin DIP 0,6 sehingga memudahkan pemasangannya.



Gambar 3.4 Pinout Sensor SHT11 dan DHT22

Dari beberapa sensor suhu dan kelembaban yang telah dijelaskan seperti contoh gambar 3.4 mulai dari spesifikasi serta kelebihan dan kekurangan masing – masing. Dari *type* SHT, memang memiliki spesifikasi dan cara kerja lebih mumpuni daripada *type* DHT dan juga dari segi fisik, *type* SHT lebih fleksibel karena lebih kecil sehingga tidak memakan tempat dan lebih efisien. Tetapi, harga yang dipatok dipasaran untuk *type* SHT lebih mahal serta jarang ditemui di indonesia, karena memang *type* SHT sebagian besar diproduksi di luar negeri, sehingga *type* DHT lah yang digunakan sebagai alternatif yang diterapkan pada alat – alat tertentu, karena selain mudah ditemui dipasaran, *type* DHT ini memiliki harga lebih terjangkau dibandingkan dengan *type* SHT meskipun memiliki fisik yang sedikit lebih besar daripada *type* SHT.

- Arduino Mega 2560

Dari jenis – jenis Arduino, Arduino memiliki banyak *type* yang mempunyai sedikit perbedaan antara satu sama lain, mulai

dari jumlah pinout untuk I/O (*Input* dan *Output*) Digital dan pinout untuk I/O Analog, serta dari segi fisik atau dimensi dan jenis input ke sumber atau koneksi yang digunakan untuk terhubung ke PC.

Jenis – jenis Arduino yang ada sekarang ini yaitu seperti Arduino *Uno*, Arduino *Due*, Arduino *Mega*, Arduino *Leonardo*, Arduino *Fio*, Arduino *Lilypad*, Arduino *Nano*, Arduino *Mini*, Arduino *Micro*, Arduino *Ethernet*, Arduino *Esplora*, dan Arduino *Robot* yang memiliki kelebihan dan kekurangan masing – masing.

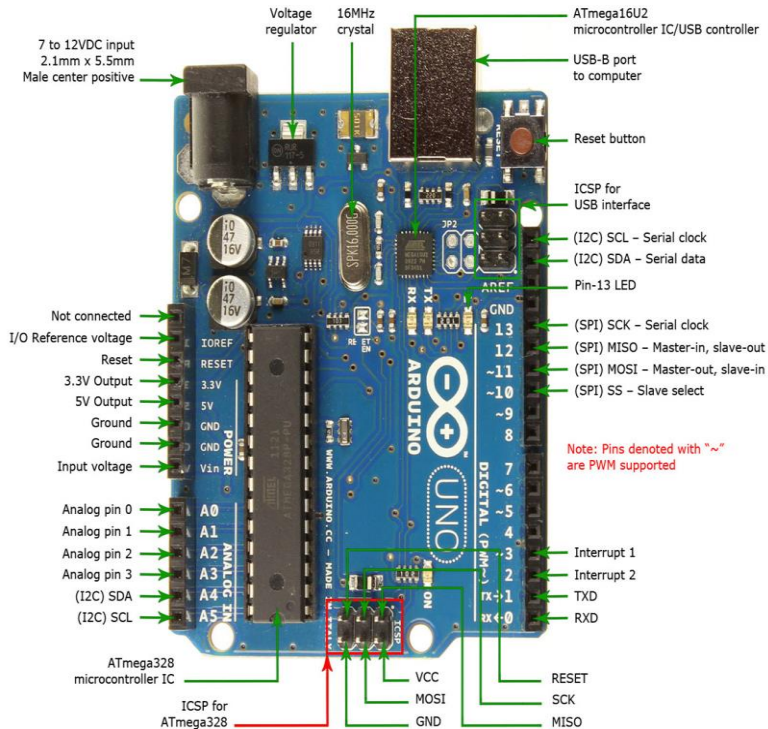
Dari sekian banyak jenis – jenis Arduino tersebut, hanya beberapa yang paling sering ditemui terutama dikalangan mahasiswa atau di indonesia untuk diterapkan pada alat – alat canggih yang membutuhkan proses berulang – ulang dan otomatis seperti robot, *conveyor*, dan sistem – sistem kontrol atau proyek – proyek tertentu. Seperti dari masing – masing typenya seperti Arduino *Uno R3*, Arduino *Nano 3*, Arduino *Pro Mini*, dan Arduino *Mega 2560* yang memiliki beberapa persamaan maupun perbedaan atau seperti dalam tabel 3.1 berikut ini.

Tabel 3.1 Spesifikasi Dari Tipe-Tipe Arduino

Deskripsi	Arduino Uno R3	Arduino Nano 3	Arduino Pro Mini	Arduino Mega 2560
Chip Mikrokontroler	ATMega328P	ATMega328P	ATMega328P	ATMega 2560
Tegangan Operasi	5V	5V	5V atau 3.3V tergantung model	
Tegangan <i>Input</i> (Rekomendasi, via Jack-DC)	7V – 12V	-	-	7V – 12V
Tegangan <i>Input (Limit, via Jack-DC)</i>	6V – 20V	-	-	6V – 20V
Digital I/O	14 Buah, 6	14 Buah, 6	14 Buah, 6	54 Buah,

Pin	diantaranya PWM	diantaranya PWM	diantaranya PWM	6 diantaran ya PWM
Analog <i>Input</i> Pin	6 Buah	6 Buah	6 Buah	16 Buah
Arus DC per Pin <i>I/O</i>	40 mA	20 mA	40 mA	20 mA
Arus DC pin 3.3V	50 mA	-	-	50 mA
Memori <i>Flash</i>	32 KB, 0.5 KB digunakan <i>bootloader</i>	32 KB, 0.5 KB digunakan <i>bootloader</i>	32 KB, 0.5 KB digunakan <i>bootloader</i>	256 KB, 8KB digunaka n <i>bootload er</i>
SRAM	2 KB	2 KB	2 KB	8 KB
EEPROM	1 KB	1 KB	1 KB	4 KB
<i>Clock Speed</i>	16 Mhz	16 Mhz	16 Mhz	16 Mhz
Dimensi	68.6 mm x 53.4 mm	45 mm x 18 mm	33 mm x 18mm	101.5 mm x 53.4 mm
Berat	25 g	5 g	5 g	37 g

Jika memperhatikan tabel 3.1 perbandingan dari ke empat Arduino tersebut, maka bisa diketahui bahwa secara spesifikasi teknis tidak terlalu banyak perbedaan antara Arduino *Uno R3*, Arduino *Nano*, dan Arduino *Pro Mini*. Ketiga nya memiliki chip *micocontroller* yang sama (ATmega328), jumlah pin yang sama, *clock speed* yang sama, dan jumlah memori yang sama. Jadi dari sudut pandang “tenaga”, ketiga board tersebut boleh di bilang sama saja.

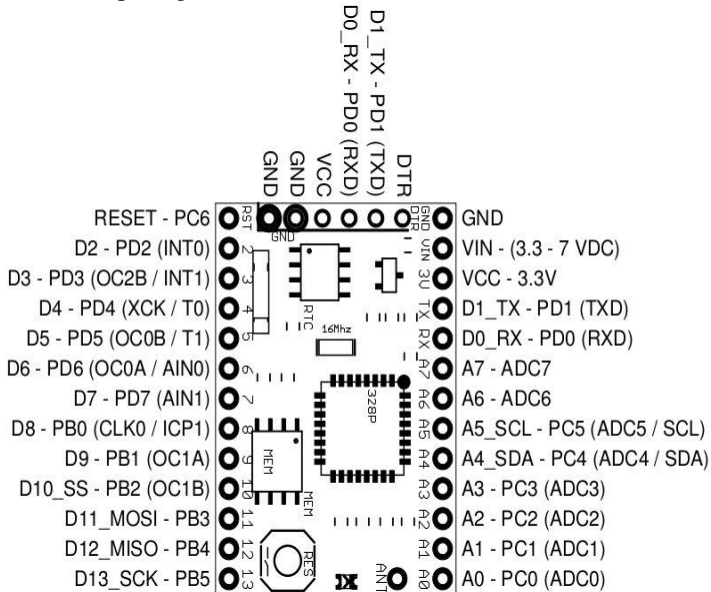


Gambar 3.5 Pinout Arduino Uno

Jadi apakah yang menjadi perbedaan antara Arduino *Uno* R3, *Nano*, dan *Pro Mini* jika secara *utility* ketiganya sama? Perbedaan utamanya terletak pada ketersediaan koneksi USB dan ketersediaan Jack-DC. Arduino *Uno* dan Arduino *Nano* sama – sama memiliki koneksi USB, sementara Arduino *Pro Mini* tidak dilengkapi chip untuk koneksi USB ke komputer, sehingga harus menggunakan board FTDI atau USB to Serial. Arduino *Uno* memiliki Jack-DC, sedangkan untuk Arduino *Nano* tidak seperti gambar 3.5.

Perbedaan lain dari ketiga board tersebut adalah kemudahan untuk koneksi kabel *jumper*. Hanya Arduino *Uno* yang memudahkan penggunaannya untuk dapat langsung mencobanya dalam melakukan eksperimen mikrokontroler,

hanya menggunakan kabel *jumper* ke *female* terminal yang tersedia. Hanya Arduino *Uno* juga dari ketiga board tersebut yang memungkinkan untuk menggunakan *shield*. Arduino *Nano* dan *Pro Mini* tidak bisa menggunakan *shield* secara langsung. Untuk mulai mencoba Arduino *Nano* dan *Pro Mini*, pengguna minimal seharusnya telah memiliki breadboard dan kabel *jumper* seperti terlihat pada gambar 3.6.

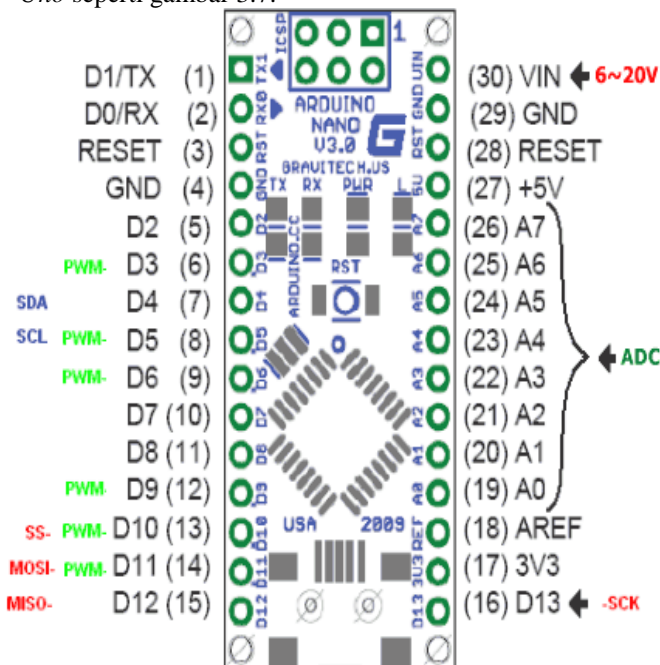


Gambar 3.6 Pinout Arduino Mini

Lalu bagaimana dengan Arduino Mega 2560?. Jadi board yang satu ini memiliki semua kelebihan dari Arduino. *Dev Board* Arduino Mega 2560 memiliki jumlah pin yang paling banyak dari kesemua board arduino, memiliki seluruh fasilitas yang ada, dan memiliki memori yang paling besar. Atau bisa dibilang Arduino Mega 2560 adalah yang paling lengkap.

Bahan pertimbangan selanjutnya dalam pemilihan Arduino yaitu dari segi ukuran atau dimensi dan kebutuhan pengguna untuk proyek apa yang akan dikerjakan. Arduino Mega 2560 memang memiliki semua apa yang dibutuhkan oleh

penggunanya, tetapi apakah harus membutuhkan sebanyak 54 pin I/O untuk kebutuhan eksperimen atau proyek yang akan dikerjakan, mengingat harga Arduino Mega 2560 adalah yang paling mahal diantara ke – empat board Arduino. Dengan pin sebanyak itu, Arduino Mega memiliki ukuran yang paling besar dari semua board. Ukuran board juga dapat menjadi penentu sebesar apa ukuran box eksperimen atau proyek yang akan dikerjakan oleh sipengguna Arduino Mega 2560 tersebut. Jika seseorang sedang berencana untuk membuat sebuah project elektronika yang berukuran kecil, tentunya Arduino *Nano* atau *Pro Mini* adalah pilihan yang tepat dibanding Arduino Mega dan *Uno* seperti gambar 3.7.



Gambar 3.7 Pinout Arduino Nano

Dari penjelasan tersebut, maka dapat ditarik kesimpulan yaitu Arduino Mega 2560 lah yang bisa dikatakan “sempurna” dari segi fisik maupun spesifikasinya untuk diterapkan dalam *prototype* rancangan alat ini.

- LCD 16x2

Display LCD merupakan sebuah *liquid crystal* atau perangkat elektronik yang dapat digunakan untuk menampilkan angka atau teks. Ada dua jenis utama layar LCD yang dapat menampilkan numerik (digunakan dalam jam tangan, kalkulator, dll) dan menampilkan teks *alfanumerik* (sering digunakan pada mesin fotokopi dan telepon genggam).

Dalam menampilkan numerik ini, kristal yang dibentuk menjadi bar, dan dalam menampilkan *alfanumerik* kristal hanya diatur kedalam pola titik. Setiap kristal memiliki sambungan listrik individu sehingga dapat dikontrol secara independen. Ketika kristal *off* (yakni tidak ada arus yang melalui kristal), maka cahaya kristal terlihat sama dengan bahan latar belakangnya, sehingga kristal tidak dapat terlihat. Namun ketika arus listrik melewati kristal, itu akan merubah bentuk dan menyerap lebih banyak cahaya. Hal ini membuat kristal terlihat lebih gelap dari penglihatan mata manusia, sehingga bentuk titik atau bar dapat dilihat dari perbedaan latar belakang.

Sangat penting untuk menyadari perbedaan antara layar LCD dan layar LED. Sebuah LED *display* (sering digunakan dalam radio dan jam) yang terdiri dari sejumlah LED yang benar – benar mengeluarkan cahaya (dan dapat dilihat dalam gelap). Sebuah layar LCD hanya mencerminkan cahaya, sehingga tidak dapat dilihat dalam gelap.

Fungsi dari *display* sendiri dalam suatu aplikasi *microcontroller* sangat penting sekali, yang diantaranya untuk :

1. Memastikan data atau program yang telah diinputkan sudah valid (benar) atau tidak.
2. Mengetahui hasil dari suatu proses atau projek.
3. Memonitoring suatu proses atau sistem
4. Mendebug program.
5. Menampilkan pesan yang diinginkan atau diprogram sebagai hasil interface dengan alat atau komponen lain.
6. Dan juga hal lainnya yang memiliki proses mekanikal atau elektrikal, terutama elektronika.

Dan jenis atau type dari LCD yang digunakan untuk projek atau aplikasi *microcontroller* tersedia berbagai ukuran seperti 2x8, 2x20, 4x20, dan 4x40 dan semuanya memiliki spesifikasi yang hampir sama, serta kelebihan dan kekurangan satu sama lain

Dari berbagai macam *type* LCD tersebut, hanya terdapat sedikit perbedaan antara satu sama lain, seperti jumlah karakter yang akan ditampilkan. Contohnya LCD 16x2 yang akan menampilkan karakternya dengan dimensi 16 untuk kolomnya dan 2 untuk barisnya yang dimana karakter – karakter yang akan ditampilkan bisa disesuaikan dengan program yang diinputkan atau seperti pada gambar 3.8.

Upper 4 Bits Lower 4 Bits	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111	
xxxx0000	CG RAM (1)			0	1	P	`	P				-	9	3	α	p	
xxxx0001	(2)			!	1	A	Q	a	q			。	ア	チ	△	ä	q
xxxx0010	(3)			"	2	B	R	b	r			「	イ	ツ	×	ß	θ
xxxx0011	(4)			#	3	C	S	c	s			」	ウ	テ	モ	ε	∞
xxxx0100	(5)			\$	4	D	T	d	t			、	エ	ト	フ	μ	Ω
xxxx0101	(6)			%	5	E	U	e	u			・	オ	ナ	1	σ	Ü
xxxx0110	(7)			&	6	F	V	f	v			ヲ	カ	ニ	ヨ	ρ	Σ
xxxx0111	(8)			'	7	G	W	g	w			ア	キ	ヌ	ラ	g	π
xxxx1000	(1)			(8	H	X	h	x			イ	ク	ネ	リ	フ	⌘
xxxx1001	(2))	9	I	Y	i	y			ウ	ケ	ル	ル	"	y
xxxx1010	(3)			*	:	J	Z	j	z			エ	コ	ハ	レ	j	〒
xxxx1011	(4)			+	;	K	[k	[オ	サ	ヒ	ロ	*	斤
xxxx1100	(5)			,	<	L	¥	l	¥			ヤ	シ	フ	ワ	φ	円
xxxx1101	(6)			-	=	M]	m]			ユ	ズ	ヘ	ン	も	÷
xxxx1110	(7)			.	>	N	^	n	^			ヨ	セ	ホ	°	ñ	
xxxx1111	(8)			/	?	O	_	o	_			ッ	ツ	マ	°	ö	■

Gambar 3.8 Karakter Pada LCD

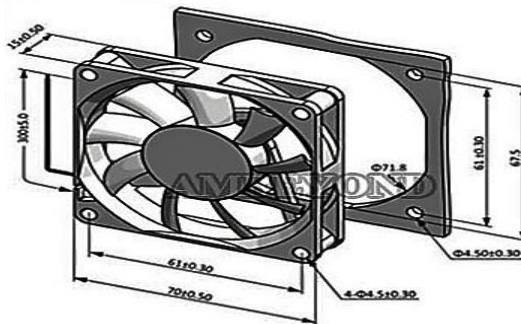
Jadi sebenarnya semua *type* LCD memiliki spesifikasi yang sebagian hampir sama, hanya saja dimensi atau jumlah keseluruhan karakter yang akan ditampilkan, mulai dari kolom dan baris. Dengan itu, pada *prototype* rancangan alat ini,

memakai LCD berukuran 16x2 karena selain harga yang dipatok lebih terjangkau, spesifikasinya juga cukup mumpuni yang sesuai dengan kebutuhan untuk menampilkan data dari *input* dan *output* pada *prototype*.

- Kipas Motor DC 5V

Sebenarnya semua Kipas Motor DC 5V memiliki spesifikasi, fungsi, maupun cara kerja yang hampir sama. Hanya saja yang membedakan dari semua tipenya yaitu dimensinya atau ukuran keseluruhan dari wadah dari kipas dan juga kipas itu sendiri. *Type* atau dimensi dari kipas motor dc 5V yaitu 20x20x6mm, 25x25x10mm, 40x40x10mm, 50x50x10mm, 60x60x25mm, 70x70x15mm, dan 80x80x25mm.

Motor DC sendiri merupakan motor listrik yang memerlukan suplai tegangan arus searah pada kumparan medan untuk diubah menjadi energi gerak mekanik. Sedangkan pada *prototype* rancangan alat ini, kipas motor dc 5V hanya digunakan untuk menjaga sirkulasi udara didalam “ruang” *prototype* tersebut, dengan cara mengganti atau membuang udara kotor didalam “ruang” *prototype* yang tentunya mengurangi nilai kelembaban yang dapat menyebabkan korosi dengan udara bersih dari luar. Maka dari itu dibutuhkan kipas motor dc yang berukuran besar atau bisa dibilang cukup agar sesuai dengan kondisi atau kebutuhan dari *prototype* rancangan alat dan juga memiliki sumber 5V yang dimana bisa terhubung ke rangkaian Arduino, dan dimensi kipas yang sesuai yaitu 70x70x15mm seperti gambar 3.9.



Gambar 3.9 Gambar Visual Kipas Motor DC 5V

- Relay DC 5V

Relay adalah Saklar (*Switch*) yang dioperasikan secara listrik dan merupakan komponen *Electromechanical* (Elektromekanikal) yang terdiri dari 2 bagian utama yakni Elektromagnet (*Coil*) dan Mekanikal (seperangkat Kontak Saklar / *Switch*). Relay menggunakan Prinsip Elektromagnetik untuk menggerakkan Kontak Saklar sehingga dengan arus listrik yang kecil (*low power*) dapat menghantarkan listrik yang bertegangan lebih tinggi.

Seperti saklar, relay juga dibedakan berdasar *pole* dan *throw* yang dimilikinya.

1. *Pole* : banyaknya *contact* yang dimiliki oleh relay
2. *Throw* : banyaknya kondisi (state) yang mungkin dimiliki *contact*

Dan berikut ini penggolongan relay berdasar jumlah *pole* dan *throw* :

1. DPST (*Double Pole Single Throw*)
2. SPST (*Single Pole Single Throw*)
3. SPDT (*Single Pole Double Throw*)
4. DPDT (*Double Pole Double Throw*)
5. 3PDT (*Three Pole Double Throw*)
6. 4PDT (*Four Pole Double Throw*)

Jenis – jenis relay :

1. *Timing* relay adalah jenis relay yang khusus. Cara kerjanya ialah sebagai berikut : jika *coil* dari timing relay ON, maka beberapa detik kemudian, baru *contact* relay akan ON atau OFF (sesuai jenis NO / NC *contact*).
2. *Latching* relay ialah jenis relay digunakan untuk *latching* atau mempertahankan kondisi aktif input sekalipun *input* sebenarnya sudah mati. Cara kerjanya ialah sebagai berikut : jika *latch coil* diaktifkan, ia tidak akan bisa dimatikan kecuali *unlatch coil* diaktifkan.

Jadi dari sekian banyak *type* atau jenis relay, yang prinsip kerja dari semua jenis juga hampir sama. Hanya saja dari sekian banyak jenis relay, Relay DC 5V lah yang paling sering dijumpai atau mudah dicari terutama diindonesia, karena pada umumnya projek elektronika menggunakan komponen relay 5v sebagai pelengkap rangkaiannya, selain fungsinya cukup membantu,

harganya juga cukup terjangkau atau lebih murah dibandingkan kenis relay yang lain.

- LED

LED atau singkatan dari *Light Emitting Diode* adalah salah satu komponen elektronik yang tidak asing lagi di kehidupan manusia saat ini. Pada dasarnya LED itu merupakan komponen elektronika yang terbuat dari bahan semi konduktor jenis dioda yang mampu memancarkan cahaya. LED merupakan produk temuan lain setelah dioda. Strukturnya juga sama dengan dioda, tetapi belakangan ditemukan bahwa elektron yang menerjang sambungan P-N. Untuk mendapatkna emisi cahaya pada semikonduktor, *doping* yang pakai adalah galium, arsenic dan phosporus. Jenis *doping* yang berbeda menghasilkan warna cahaya yang berbeda pula.

- Kelebihan dari LED :

- LED memiliki efisiensi energi yang lebih tinggi dibandingkan dengan lampu lain, dimana LED lebih hemat energi 80 % sampai 90% dibandingkan lampu lain.
- LED memiliki waktu penggunaan yang lebih lama hingga mencapai 100 ribu jam.
- LED memiliki tegangan operasi DC yang rendah.
- Cahaya keluaran dari LED bersifat dingin atau cool (tidak ada sinar UV atau energi panas).
- Ukurannya yang mini dan praktis

- Kelemahan LED

- Suhu lingkungan yang terlalu tinggi dapat menyebabkan gangguan elektrik pada LED.
- Harga LED per lumen lebih tinggi dibandingkan dengan lampu lain.
- Kelemahan dari LED di atas yang menyebabkan masyarakat lebih memilih menggunakan Cara penerangan biasa dengan lampu pijar maupun neon dibandingkan menggunakan

- 7 Segment

Common katode adalah semua pin katoda pada 7 *segments* disatukan, sedangkan pin anoda dihubungkan ke *port-port* pada mikokontroller. Common katoda digunakan pada rangkaian yang memerlukan aktif tinggi (*active high*).

Common Anode adalah LED pada 7 *segment* semua pin anode-nya dihubungkan menjadi satu, sedangkan pin katoda dihubungkan ke *port-port* pada mikrokontroller. Common anode digunakan untuk rangkaian yang memerlukan aktif rendah (*active low*).

Driver seven segmen bisa berupa IC 7447, IC 7448, IC Latch 74HC573, ULN2803, mikrokontroller, dll. Setiap penggunaan driver tersebut masing-masing memiliki kelebihan dan kekurangan, antara lain:

- Penggunaan IC7448 atau IC7447 yaitu decoder BCD ke seven segmen bagus digunakan untuk tampilan seven segmen 4 digit karena hanya akan menggunakan 1 buah port mikrokontroller saja akan tetapi tampilan hanya berupa angka dan tidak dapat tampilan lainnya.
- Penggunaan mikrokontroller bagus untuk sistem scanning seven segmen hingga 8 digit atau 24 digit dengan tampilan berupa angka atau karakter-karakter tertentu dan sangat fleksibel selain itu rangkaian menjadi lebih sederhana karena hanya membutuhkan resistor dan transistor saja. Tetapi karena menggunakan sistem scanning maka untuk program yang panjang dan apabila pada program tersebut terdapat penggunaan delay yang cukup lama maka seven segmen akan mati selama delay tersebut karena Seven segmen tidak discanning.
- Penggunaan IC Latch 74HC573 atau IC latch lainnya bagus untuk sistem scanning dengan tampilan berupa angka atau karakter-karakter tertentu dan sangat fleksibel dengan jumlah digit yang banyak (misal 8 digit, 16 digit, 24 digit, atau lebih). Karena menggabungkan sistem scanning dan Latch maka untuk program yang panjang atau terdapat delay sekalipun tidak akan berpengaruh pada tampilan seven segmennya, akan tetapi karena menggunakan IC latch tambahan yang memperbesar rangkaian keseluruhan.
- Penggunaan IC ULN2803 sama seperti pada penggunaan mikrokontroller dengan sistem scanning akan tetapi arus dan tegangannya bisa lebih besar untuk mengendalikan seven segmen yang membutuhkan arus atau tegangan besar.Selenoid *Door lock* 12V

- Selenoid *Door Lock* 12V

Solenoid memiliki 2 jenis, yaitu solenoid *valve* dan solenoid *door lock*. Solenoid *valve* merupakan solenoid atau sebuah katup yang dioperasikan secara elektromekanik. Katup dikendalikan oleh arus listrik melalui solenoid: dalam kasus katup dua port, aliran dinyalakan atau dimatikan; Dalam kasus katup tiga port, arus keluar akan dialihkan di antara dua port *output*. Beberapa katup solenoida dapat ditempatkan bersamaan pada bermacam – macam jenis pipa.

Katup solenoid adalah bagian atau part kontrol yang paling sering digunakan dalam bidang perairan atau sejenisnya. Fungsi dari solenoid *valve* ini adalah mematikan, melepaskan, mengatur jumlah debit air yang dibutuhkan, mendistribusikan atau mencampur cairan. Hal ini dapat ditemukan di banyak proyek – proyek yang berhubungan. Solenoida dapat membantu dengan cara perpindahan yang cepat dan aman, kehandalan yang tinggi, umur pemakaian yang panjang, kompatibilitas media yang baik dari bahan yang digunakan, daya kontrol rendah dan desain yang terpadu.

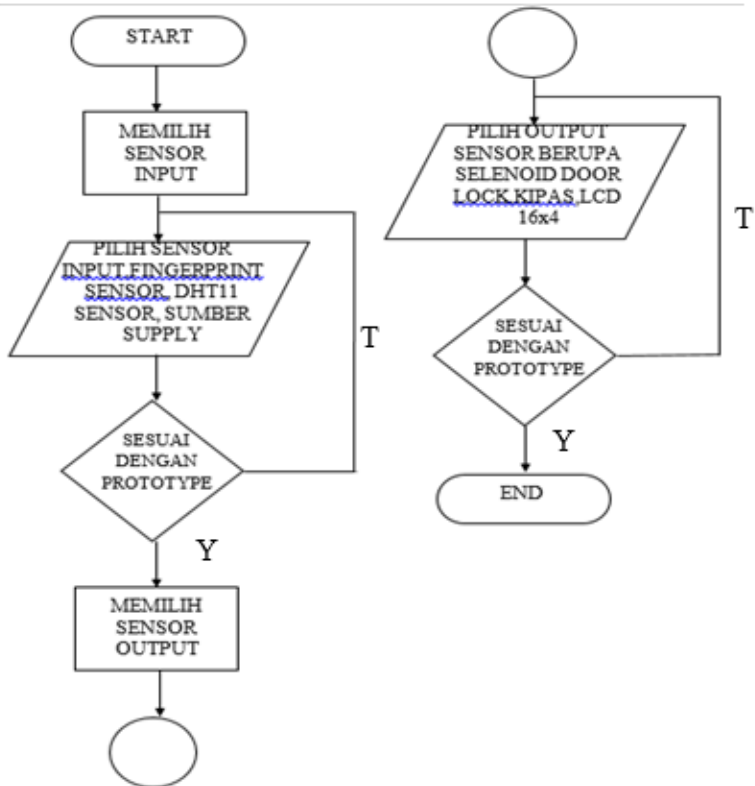
Selain aktuator yang bertipe penyelam (jenis solenoid *valve* yang terdapat didalam air) yang paling sering digunakan, aktuator rotor dan aktuator yang berputar juga digunakan.

Solenoid *door lock*, atau solenoid yang umumnya digunakan sebagai pengaman dalam konstruksi sebuah bangunan (rumah, hotel, gerbang, dan sejenisnya), kendaraan (mobil, motor, dan sejenisnya), ataupun proyek – proyek yang lain yang memungkinkan pengelolaan akses kontrol secara lebih terpadu, lebih aman, dan lebih efisien.

Dalam prototype ini menggunakan solenoid berjenis *door lock* yang membutuhkan suplai tegangan 12V. Karena selain mudah didapatkan dan memiliki harga terjangkau, solenoid *door lock* hanya ada 2 jenis yaitu solenoid *door lock* 12V dan solenoid *door lock* 24V yang memiliki banyak kesamaan.

3.3 Flowchart Diagram

Dalam sub bab ini akan dijelaskan alur proses perancangan alat.



Gambar 3.10 Flowchart Proses Desain Kebutuhan Platform

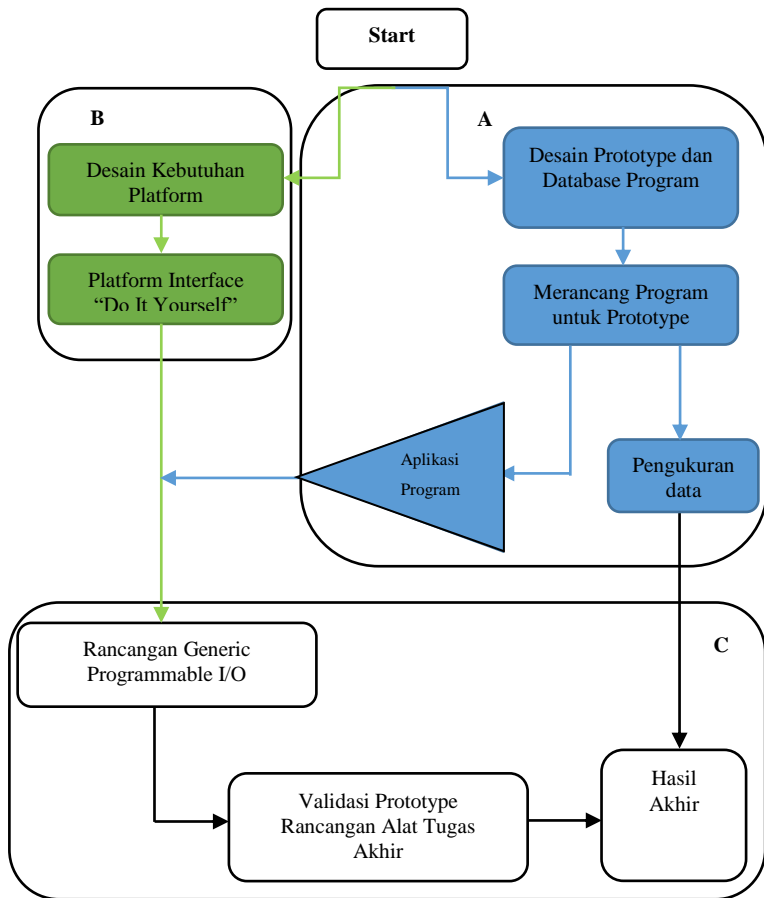
keterangan :

Pada gambar 3.10 flowchart di atas dijelaskan proses pemilihan sensor *input* dan *output* sesuai atau tidak dengan *prototype* apabila tidak sesuai maka proses akan memulai dari awal sebagai contoh saja di dalam *flowchart* di atas di pilih sensor acak agar sistem dapat berjalan sesuai dengan memilih beberapa sensor *input* dan *output* agar sebuah sistem dapat di tampilkan dan ada *feedbacknya*. Selain itu *flowchart* di atas di buat dengan tujuan agar dalam merancang *software* 'Do It By Yourself' mudah dan tidak mengalami kesulitan. Bisa di bilang *flowchart* di atas sebagai awal mula pondasi dalam membuat *software*

dengan menggunakan VB. Proses di atas hanya untuk sensor *input* dan *output* karena sensor *input* dan *output* memerlukan konfigurasi yang rumit dan juga harus memilih tipe yang cocok dengan Arduino yang akan di pakai juga harus menyesuaikan dan memilih tipe *library* yang cocok dengan Arduino yang akan di pakai, karena *library* yang tersedia di internet sangat banyak dan juga sangat beragam, belum tentu juga cocok dengan sensor *input* dan *output* yang akan di pilih.

3.4 Pembagian Tugas Pekerjaan

Dalam mengerjakan Tugas Akhir ini pembagian tugas untuk masing – masing orang per kelompok dibagi menjadi beberapa rincian seperti yang akan kami jelaskan seperti pada alur pengerjaan di bawah ini, dimana proses pengerjaan dibagi menjadi 2 blok (sesuai dengan gambar blok diagram pada Bab I) yaitu blok “A”, blok “B”, dan Blok “C”. Dimana blok pertama atau blok “A” akan dibagi tugaskan kepada Rendra Kurnia .R sebagai mahasiwa Pertama. Pada area blok “A” yang diartikan pada awalnya membuat *prototype* rancangan alat Tugas Akhir dan *database* untuk keseluruhan program dari input dan output yang akan digunakan, lalu langkah selanjutnya pembagian program yang didapatkan dari *database* yang disesuaikan dengan kebutuhan dari user. Setelah itu dibagi menjadi dua langkah yaitu pengukuran data awal dari program yang telah diambil dari *database* program untuk *input* dan *output* yang dibutuhkan oleh *user* untuk diproses dan mengevaluasi hasil dari data yang telah diambil agar program menjadi cepat dan tepat. Dan langkah lainnya yaitu aplikasi *software* dari beberapa program yang diambil dari *database* dapat digunakan sesuai *interface* yang dibutuhkan *user*. Kemudian blok yang kedua atau pada area blok “B” yang diberikan tugas kepada Setyo Budi Utomo sebagai mahasiwa kedua yang membuat Desain kebutuhan platform yang dimaksudkan yaitu desain *hardware*, yang dimana dapat menjalankan sistem atau program agar dapat berjalan sesuai dengan rancangan. Selanjutnya *Platform Interface* “*Do It by Yourself*” dimana dari hal ini dibutuhkan sistem yang dapat dijalankan sesuai dengan kebutuhan *user* tanpa harus mengetahui secara detail bagaimana alur dari jalannya program yang dibutuhkan.



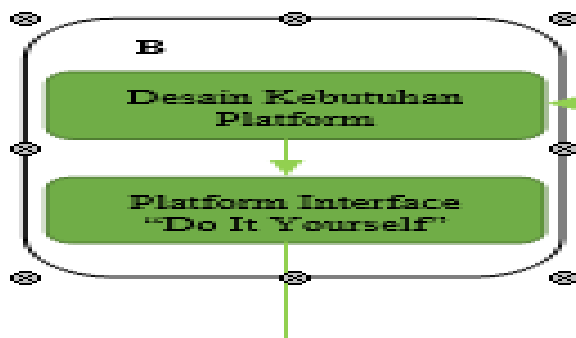
Gambar 3.11 Flowchart Pembagian Tugas Kelompok

Lalu dari kedua bagian tersebut digabungkan jadi satu dalam area blok "C" akan menjadi sebuah *Generic Programmable I/O*. Dimana pada bagian ini setelah *software "Do It by Yourself"* yang telah di terapkan dalam memprogram *prototype* secara otomatis dengan *input* dan *output* yang di tentukan oleh *user* itu sendiri. Selanjutnya ada validasi *prototype* rancangan alat Tugas Akhir yang dimana hasil evaluasi dari proses sebelumnya agar rancangan alat lebih sempurna dan mempunyai data yang tepat sesuai dengan kebutuhan *user* dari

rancangan alat. Dan yang terakhir terdapat bagian hasil akhir yang dimana hasil yang sudah tepat dari *platform* maupun program yang dibutuhkan oleh *user* dan sesuai dengan data yang telah dievaluasi terus menerus hingga terverifikasi dengan benar.

Rincian pengerjaan masing – masing mahasiswa akan lebih rinci dijelaskan pada sub bab di bawah ini

3.4.1 Proses Pengerjaan Blok “B”



Gambar 3.12 Proses Pengerjaan Alur B

Pada proses pengerjaan alur b masing-masing proses memiliki digram *flowchart* masing-masing yang akan di jelaskan menurut alur kerjanya dan pada masing-masing proses tersebut di buat *flowchart* gunanya ialah agar proses pengerjaan lebih mudah untuk di pahami

Pada area Blok “B” proses pengerjaan dimulai dengan mencari *hardware* (sensor, actuator, dan kontroler) kebutuhan dari *Prototype*, dimana *hardware* (sensor, actuator, dan kontroler) akan di diterapkan pada dua objek. Yang pertama adalah pada *Generic Programmable* penghubung pengguna dengan I/O dari Arduino dimana *hardware* yang di butuhkan diantaranya adalah :

A. Sensor *Fingerprint Optical* R305

Sensor ini digunakan sebagai pembaca data pola sidik jari yang sebelumnya telah di *inputkan* maupun yang akan di *inputkan* dan disimpan sebagai data yang telah dikenali. Yang nantinya berfungsi agar hanya orang – orang tertentu

yang dapat mengakses *box* pengaman perangkat BTS (*Base Transceiver Station*)

B. Sensor DHT11

Sensor ini sebagai Sensor yang digunakan untuk mendeteksi suhu sekaligus mengukur perubahan kelembaban, fungsinya ialah sebagai monitoring suhu dan kelembaban dalam *box* pengaman perangkat BTS (*Base Transceiver Station*) dengan tindakan apabila suhu dan kelembaban melebihi batas aman untuk komponen elektronika di dalam maka akan memberikan *output* pada kipas sehingga sirkulasi udara dapat berjalan dengan baik, mengingat tempat perangkat BTS (*Base Transceiver Station*) yang sering tempatnya terpencil. sehingga membutuhkan sistem pengaturan suhu dan kelembaban sendiri sehingga membuat kinerja komponen elektronika menjadi lebih maksimal.

C. Solenoid Door Lock

Dalam *box* pengaman perangkat BTS (*Base Transceiver Station*) ini berfungsi sebagai actuator untuk mengunci dan membuka pintu *box*. Dalam pengertian, solenoid hanya akan membuka apabila sensor *fingerprint* mengenali pola sidik jari yang sebelumnya telah diinputkan dan disimpan.

D. Power Bank

Dalam hal ini, sumber 5V di ambil dari *output* powerbank sehingga sistem dapat terus bekerja dan sumber dapat diisi ulang apabila habis, sehingga alat ini bisa juga bersifat *portable*, sehingga sistem yang telah dirancang ini dapat berjalan meskipun adanya daya turun ataupun pemadaman listrik di daerah tempat BTS (*Base Transceiver Station*) berada.

E. Arduino Mega 2560

Arduino ini dipergunakan karena selain sebagai mini prosesor yang harganya terjangkau, juga mempunyai sistem yang dapat diintegrasikan dengan *Visual Basic* sehingga dapat di atur dengan sedemikian rupa agar mempermudah *user* dalam mengatur dan menyesuainya.

F. LCD 16X2

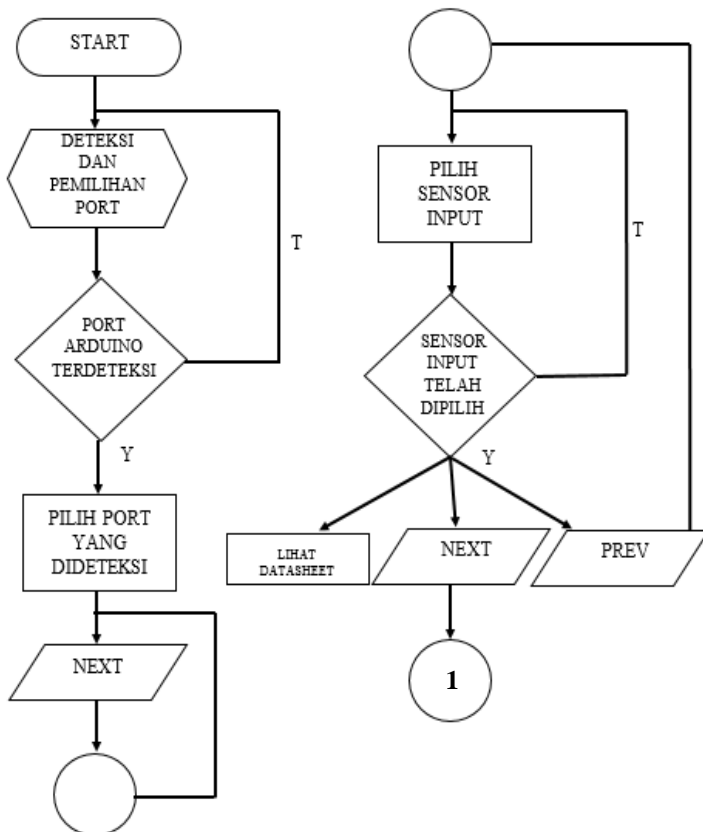
LCD 16X2 di fungsikan sebagai penampil suhu dan kelembaban yang terukur oleh sensor DHT11. Tampilan

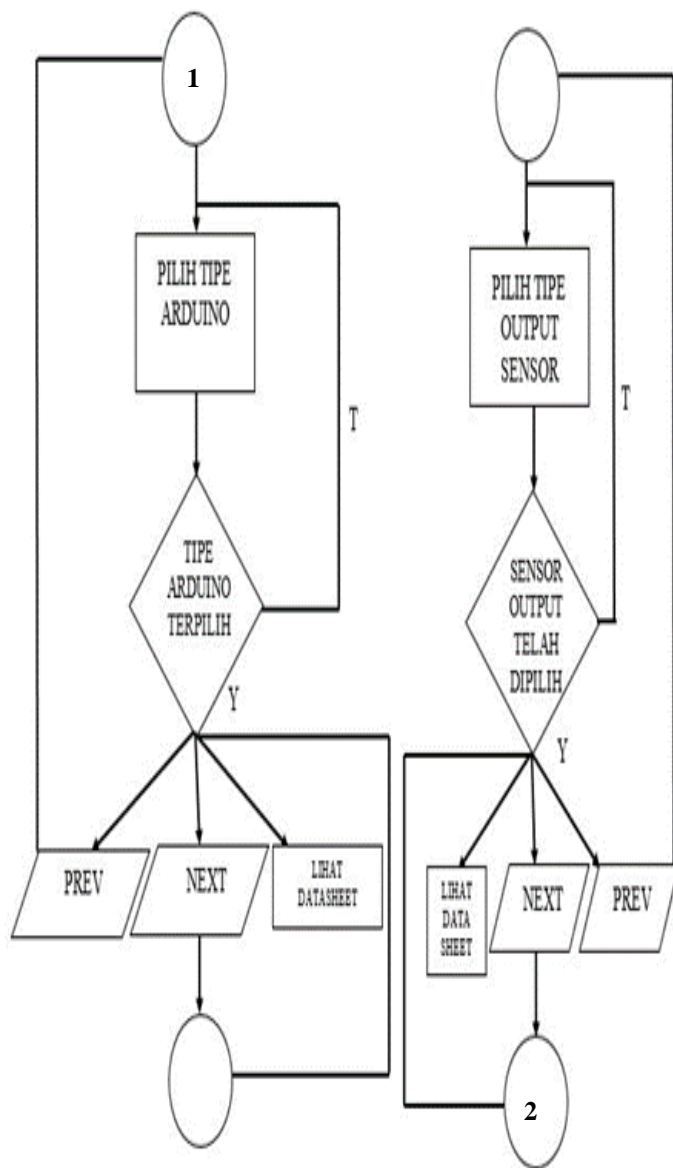
yang dimaksudkan adalah menampilkan kondisi tentang keadaan suhu dan kelembaban di dalam *box* pengaman perangkat BTS (*Base Transceiver Station*)

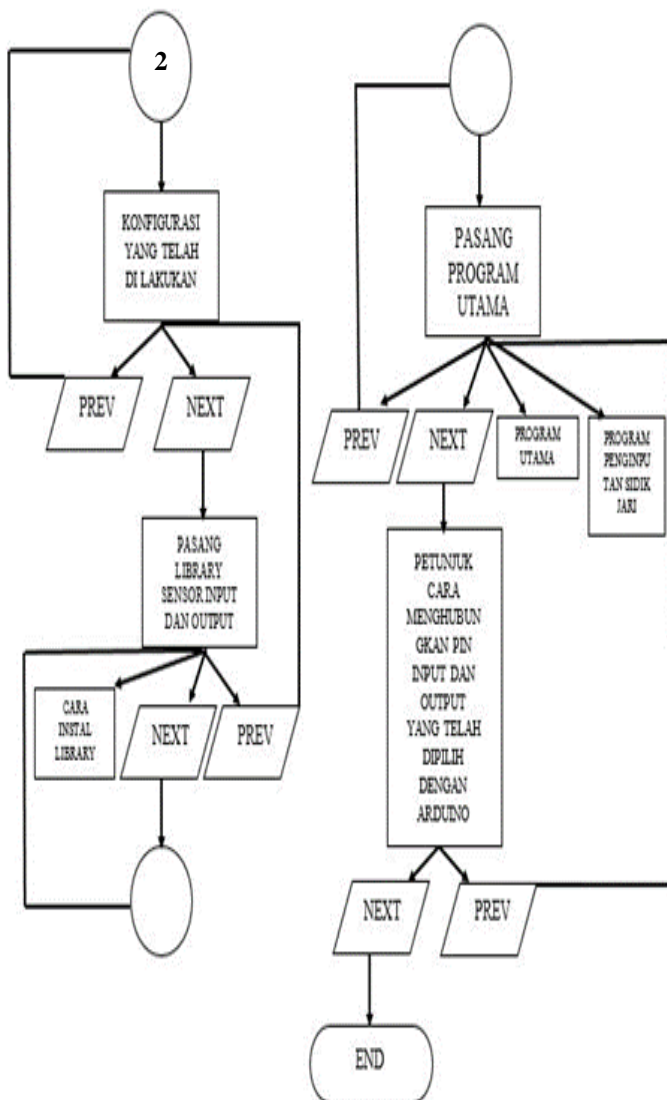
G. Kipas Motor DC 5V

Fungsi kipas sendiri dalam rancangan alat, digunakan sebagai pengatur sirkulasi udara agar suhu dan kelembaban di dalam *box* sesuai dengan batas aman suhu dan kelembaban untuk komponen elektronika.

Selanjutnya proses pengerjaan dimulai dengan Proses pembuatan *Platform Interface* “*Do It by Yourself*” seperti langkah – langkah dan gambar 3.13 *flow chart* di bawah ini.

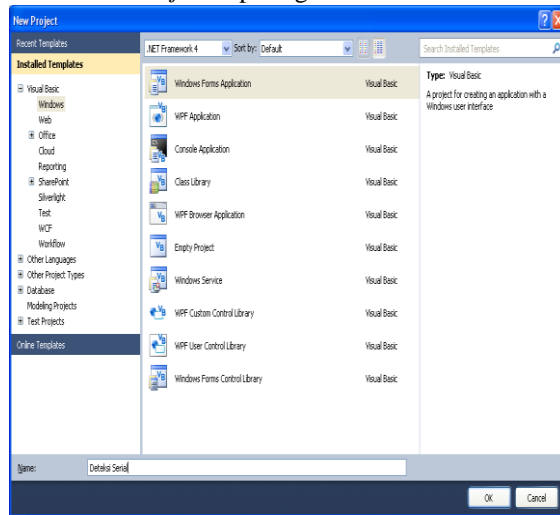






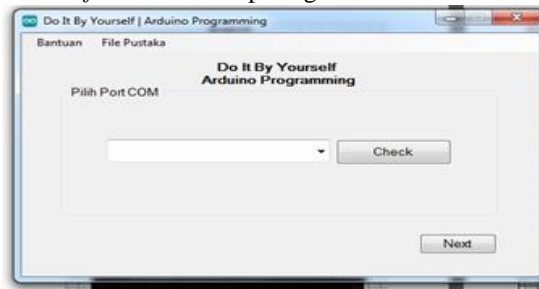
Gambar 3.13 Flowchart Perancangan Software “Do It By Yourself”

1. Pilih *New Project* seperti gambar 3.14



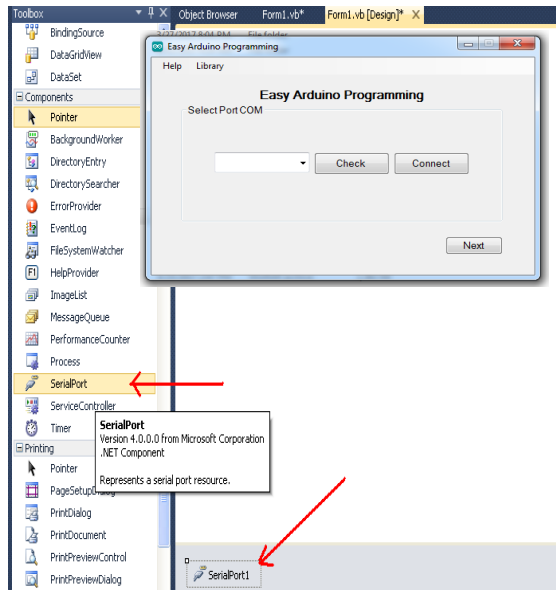
Gambar 3.14 Langkah-langkah Membuat *New Project*

2. Buat *form* berikut seperti gambar 3.15



Gambar 3.15 Form memilih *Port Arduino*

3. Isikan coding berikut seperti gambar 3.16:
-Tambahkan Komponen Serial Port dari group *Component*



Gambar 3.16 Petunjuk Mengisikan Kode Pemrograman Pada *Form*

- Pada *button 1 (Button close)* tambahkan program ini

```
Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button1.Click
    Dim Ingin As MsgBoxResult
    Ingin = MsgBox("Apakah Anda akan keluar?",
    MsgBoxStyle.YesNo, "Keluar")
    If Ingin = MsgBoxResult.Yes Then
        Me.Close()
    ElseIf Ingin = MsgBoxResult.No Then
        Exit Sub
    End If
End Sub
```
- Pada *button connect* tambahkan program seperti dibawah ini

```

Private Sub btnConnect_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs)
Handles btnConnect.Click
    'cek, jika status serial port open, tutup
    koneksi serial port
    If SerialPort1.IsOpen Then
        SerialPort1.Close()
    End If

    'atur properti serial port
    Try
        With SerialPort1
            .PortName = cbbCOMPorts.Text
            .BaudRate = 115200
            .ReadBufferSize = 500
            .Parity = IO.Ports.Parity.None
            .DataBits = 8
            .StopBits = IO.Ports.StopBits.One
            .Handshake = IO.Ports.Handshake.None
        End With
        'open koneksi serial port
        SerialPort1.Open()

        btnConnect.Enabled = False
        btnDisconnect.Enabled = True
        Catch ex As Exception
            MsgBox(ex.ToString)
        End Try
    End Sub

```

- **Untuk Coding Main menu**
Imports System.IO
Imports System.Text
Imports System

```

Public Class MainMenu

    Private Sub Button1_Click(sender As
Object, e As EventArgs)
        Label1.Text = "HELLLLLLLO"
    End Sub

```

```

        Private Sub Button2_Click(sender As
Object, e As EventArgs) Handles
Button2.Click
            Me.Visible = False
        End Sub

```

```

        Private Sub Button4_Click(sender As
Object, e As EventArgs) Handles
Button4.Click

```

```

        System.Diagnostics.Process.Start("C:\Pr
ogram Files (x86)\Arduino\arduino.exe",
"c:\xxx\esp8266_2\esp8266_2.ino")
        End Sub

```

```

        Private Sub Button5_Click(sender As
Object, e As EventArgs) Handles
Button5.Click
            Dim objFSO, strFolder
            strFolder = "C:\xxx"
            objFSO =
CreateObject("Scripting.FileSystemObjec
t")

```

```

                If Not
objFSO.FolderExists(strFolder) Then

```

```

objFSO.CreateFolder(strFolder)
                MsgBox("Success Create
Folder", MsgBoxStyle.Information,
"Create Folder")

```

```

                Else
                    MsgBox("Folder Name is
exist", MsgBoxStyle.Critical, "Error")
                End If
            End Sub

```

```

        Private Sub Button6_Click(sender As
Object, e As EventArgs) Handles
Button6.Click

```

```

        Dim path As String =
"c:\xxx\MyTest.ino"

        ' Create or overwrite the file.
        Dim fs As FileStream =
File.Create(path)
        Dim program As String

        Try
            ' Add text to the file.
            Dim info As Byte() = New
UTF8Encoding(True).GetBytes("aaaa")
            fs.Write(info, 0,
info.Length)
            fs.Close()
            MsgBox("Success Create a
File", MsgBoxStyle.Information, "Create
File")

            Catch ex As Exception
                MsgBox(ex.ToString,
MsgBoxStyle.Critical, "Error")
            End Try

        End Sub

        Private Sub Button7_Click(sender As
Object, e As EventArgs) Handles
Button7.Click
            Dim fileReader As
System.IO.StreamReader
            fileReader =
My.Computer.FileSystem.OpenTextFileRead
er("C:\xxx\esp8266_2.ino")
            Dim stringReader As String
            stringReader = ""
            Try
                While Not
fileReader.EndOfStream

```

```

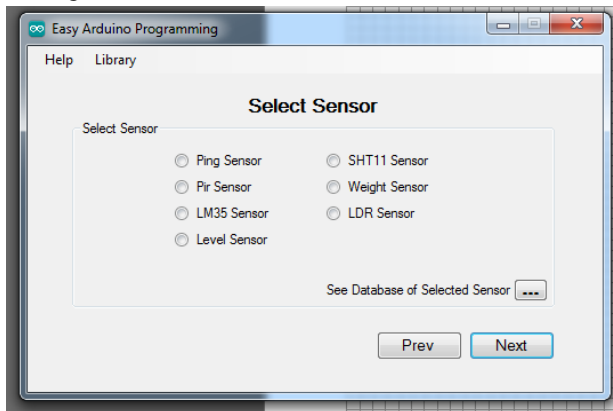
        stringReader =
stringReader & vbCrLf &
fileReader.ReadLine()
        End While

        Form2.TextBox1.Text =
stringReader
        'MsgBox("Text in the file
is :" & vbCrLf & stringReader,
MsgBoxStyle.Information, "Read File")
        Form2.Show()
        Catch ex As Exception
        MsgBox(ex.ToString,
MsgBoxStyle.Critical, "Error")
        End Try

    End Sub
End Class

```

4. Buat *form* untuk memilih sensor yang akan di pakai seperti gambar 3.17



Gambar 3.17 Membuat *Form* Untuk Memilih Sensor

5. Isikan kode berikut
Public Class SelectSensor


```

        Private Sub Button1_Click(sender As
Object, e As EventArgs) Handles
Button1.Click
            Form1.MainMenu1.Visible = True
        End Sub

```

```

        Private Sub Button2_Click(sender As
Object, e As EventArgs) Handles
Button2.Click
            selectSensor()
            If sensorSelected = 0 Then
                myMsg()
            Else
                Me.Visible = False
            End If
        End Sub

```

```

        Private Sub Button3_Click(sender As
Object, e As EventArgs) Handles
PB_ViewDBSensor.Click
            selectSensor()
            sensorSelected_f = 1
            If sensorSelected = 0 Then
                myMsg()
            Else

```

```

'MsgBox(sensorSelected.ToString)
        view_datasheet.Show()

        'Select Case sensorSelected
        ' Case 1
        ,
myMsg2("http://playground.arduino.cc/Ma
in/DHT11Lib")
        ' Case 2
        ,
myMsg2("https://learn.adafruit.com/adaf
ruit-optical-fingerprint-sensor/wiring-
for-use-with-arduino")

```

```

        ' Case 3
        ' myMsg2 ("www.LM35.com")
        ' Case 4
        '
myMsg2 ("www.LevelSensor.com")
        ' Case 5
        ' myMsg2 ("www.SHT11.com")
        ' Case 6
        '
myMsg2 ("www.weightSensor.com")
        ' Case 7
        '
myMsg2 ("www.LDRSensor.com")
        ' Case Else
        'End Select
    End If
End Sub

Private Sub
SelectSensor_Load(sender As Object, e
As EventArgs) Handles MyBase.Load
    sensorSelected = 0
    sensorSelected_f = 0
    RB_LM35.Enabled = False
    RB_LevelSensor.Enabled = False
    RB_SHT11.Enabled = False
    RB_WeightSensor.Enabled = False
    RB_LDR.Enabled = False

End Sub

Private Sub myMsg2(txt As String)
    MsgBox("Please visit : " & txt
& " to view sensor datasheet",
MsgBoxStyle.Information, "Sensor
Datasheet")
End Sub

Private Sub myMsg()

```

```

        MsgBox("You have not selected
the sensor yet! Please select the
sensor.", MsgBoxStyle.Critical, "Select
Sensor")
    End Sub

```

```

Private Sub selectSensor()
    'Get Selected Sensor
    If RB_DHT11.Checked Then
        sensorSelected = 1
    End If

    If RB_Fingerprint.Checked Then
        sensorSelected = 2
    End If

    If RB_Fingerprint.Checked And
RB_DHT11.Checked Then
        sensorSelected = 3
    End If

    If Not RB_DHT11.Checked And Not
RB_Fingerprint.Checked Then
        sensorSelected = 0
    End If

    'If RB_LM35.Checked Then
    ' sensorSelected = 3
    'ElseIf RB_LevelSensor.Checked
Then
    ' sensorSelected = 4
    'ElseIf RB_SHT11.Checked Then
    ' sensorSelected = 5
    'ElseIf RB_WeightSensor.Checked
Then
    ' sensorSelected = 6
    'ElseIf RB_LDR.Checked Then
    ' sensorSelected = 7
    'End If

```

```

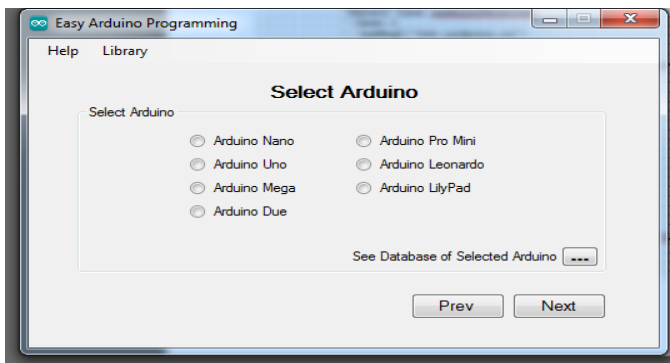
'MsgBox(sensorSelected.ToString)
End Sub

Private Sub Button3_Click_1(sender
As Object, e As EventArgs) Handles
Button3.Click
    selectSensor()
    MsgBox(sensorSelected.ToString)

End Sub
End Class

```

6. Buat *form* untuk memilih tipe Arduino yang akan dipakai seperti gambar 3.18.



Gambar 3.18 Membuat *Form* Memilih Tipe Arduino

7. Isikan kode berikut.

```

Public Class SelectArduino

    Private Sub Button1_Click(sender As
Object, e As EventArgs) Handles
Button1.Click
        Form1.SelectSensor1.Visible = True
    End Sub

```

```

        Private Sub Button2_Click(sender As
Object, e As EventArgs) Handles
Button2.Click
            selectArduino()
            If arduinoSelected = 0 Then
                myMsg()
            Else
                Me.Visible = False
                If sensorSelected = 1 And
arduinoSelected = 1 Then

                    End If
                End If
            End Sub

        Private Sub Button3_Click(sender As
Object, e As EventArgs) Handles
PB_ViewDBArduino.Click
            selectArduino()
            'Flag for Select Sensor
            sensorSelected_f = 0
            If arduinoSelected = 0 Then
                myMsg()
            Else
                view_datasheet.Show()

'MsgBox(arduinoSelected.ToString)
                'Select Case arduinoSelected
                ' Case 1
                ' myMsg2("www.arduino.cc")
                ' view_datasheet.Show()
                ' Case 2
                ' myMsg2("See at
https://www.arduino.cc/en/Main/ArduinoBoardU
no")
                ' Case 3
                ' myMsg2("www.arduino.cc")
                ' Case 4
                ' myMsg2("www.arduino.cc")
                ' Case 5
                ' myMsg2("www.arduino.cc")
                ' Case 6
                ' myMsg2("www.arduino.cc")
                ' Case 7

```

```

        ' myMsg2("www.arduino.cc")
        ' Case Else
        'End Select
    End If
End Sub

Private Sub SelectArduino_Load(sender As
Object, e As EventArgs) Handles MyBase.Load
    arduinoSelected = 0
End Sub

Private Sub myMsg2(txt As String)
    MsgBox("Please visit : " & txt & "
to view datasheet", MsgBoxStyle.Information,
"Arduino Datasheet")
End Sub

Private Sub myMsg()
    MsgBox("You have not selected the
arduino yet! Please select the arduino
type.", MsgBoxStyle.Critical, "Select
Arduino")
End Sub

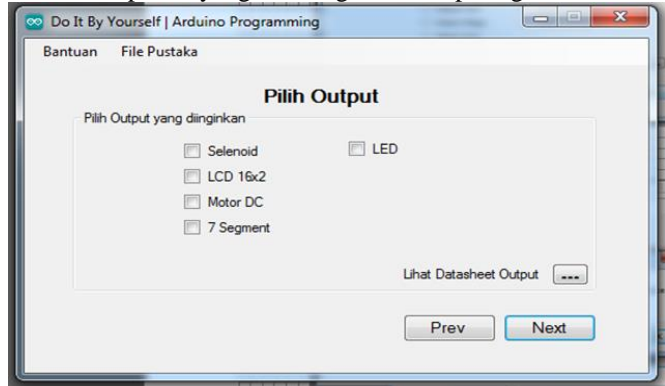
Private Sub selectArduino()
    'Get Selected Arduino
    If RB_Nano.Checked Then
        arduinoSelected = 1
    ElseIf RB_Uno.Checked Then
        arduinoSelected = 2
    ElseIf RB_Mega.Checked Then
        arduinoSelected = 3
    ElseIf RB_Due.Checked Then
        arduinoSelected = 4
    ElseIf RB_ProMini.Checked Then
        arduinoSelected = 5
    ElseIf RB_Leonardo.Checked Then
        arduinoSelected = 6
    ElseIf RB_LilyPad.Checked Then
        arduinoSelected = 7
    End If

    'MsgBox(arduinoSelected.ToString)
End Sub

```

End Class

8. Buat *form* untuk setting pin *input* dan *output* dari tipe Arduino yang telah dipilih dan akan dipakai untuk komponen yang akan digunakan seperti gambar 3.19.



Gambar 3.19 Membuat *Form* Untuk memilih *Output*

9. Isikan kode berikut

```
Public Class SettingPin

    Private Sub Button2_Click(sender As Object, e As EventArgs) Handles Button2.Click
        'checkPin()
        'If inputPin = "" Then
            ' MsgBox("You have not selected the Pins yet, Please select for Pins!", MsgBoxStyle.Critical, "Select Pin")
        'Else
            ' Me.Visible = False
        'End If
    End Sub

    Private Sub Button1_Click(sender As Object, e As EventArgs) Handles Button1.Click
        Form1.SelectArduino1.Visible = True
    End Sub
End Class
```

```

End Sub

Private Sub CB_D0_CheckedChanged(sender
As Object, e As EventArgs)

End Sub

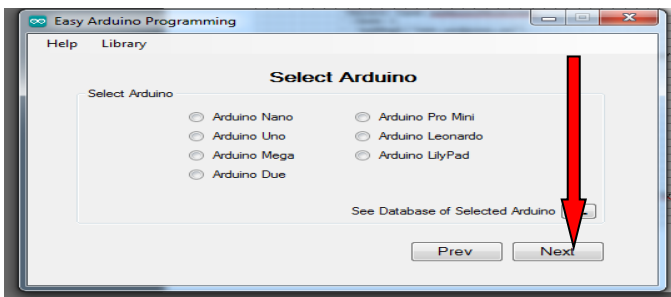
Private Sub SettingPin_Load(sender As
Object, e As EventArgs) Handles MyBase.Load

End Sub

Private Sub checkPin()
    'If CB_D0.Checked Then
    ' inputPin = inputPin & "D0#"
    'End If
    'If CB_D1.Checked Then
    ' inputPin = inputPin & "D1#"
    'End If
    'If CB_D2.Checked Then
    ' inputPin = inputPin & "D2#"
    'End If
    'If CB_D3.Checked Then
    ' inputPin = inputPin & "D3#"
    'End If
    'If CB_D4.Checked Then
    ' inputPin = inputPin & "D4#"
    'End If
End Sub
End Class

```

10. Pada message box *View Datasheet* seeperti gambar 3.20



Gambar 3.20 Membuat *Form* Melihat DataSheet Arduino

11. Isikan kode berikut

```
Public Class view_datasheet
    Dim txt As String

    Private Sub Button1_Click(sender As
Object, e As EventArgs) Handles
Button1.Click
        Me.Close()

    End Sub

    Private Sub Button2_Click(sender As
Object, e As EventArgs) Handles
Button2.Click
        Try

System.Diagnostics.Process.Start(txt)
        Catch ex As Exception

        End Try

    End Sub

    Private Sub view_datasheet_Load(sender
As Object, e As EventArgs) Handles
MyBase.Load
        Dim txt2 As String
        If sensorSelected_f = 0 Then
            txt2 = "To see datasheet of
selected arduino, please visit this link : "
& vbCrLf

            Select Case arduinoSelected
                Case 1
                    txt =
"https://www.arduino.cc/en/Main/ArduinoBoard
Nano"
                Case 2
                    txt =
"https://www.arduino.cc/en/main/arduinoBoard
Uno"
                Case 3
```

```

txt =
"https://www.arduino.cc/en/Main/arduinoBoard
Mega"
    Case 4
        txt =
"https://www.arduino.cc/en/Main/arduinoBoard
Due"
    Case 5
        txt =
"https://www.arduino.cc/en/Main/arduinoBoard
ProMini"
    Case 6
        txt =
"https://www.arduino.cc/en/main/arduinoBoard
Leonardo"
    Case 7
        txt =
"https://www.arduino.cc/en/Main/arduinoBoard
LilyPad"
    Case Else
        txt = ""
    End Select
Else
    txt2 = "To see datasheet of
selected sensor, please visit this link :" &
vbCrLf
    Select Case sensorSelected
        Case 1
            txt =
"http://playground.arduino.cc/Main/DHT11Lib"
        Case 2
            txt =
"https://learn.adafruit.com/adafruit-
optical-fingerprint-sensor/wiring-for-use-
with-arduino"
        Case 3
            txt2 = "To see datasheet
of selected sensor, please visit this link
:" & vbCrLf
            txt2 = txt2 & "For DHT11
Sensor you can view datasheet at this link
:" & vbCrLf

```

```

        txt2      =      txt2      &
"http://playground.arduino.cc/Main/DHT11Lib"
& vbCrLf & vbCrLf
        txt2      =      txt2      & "For
Fingerprint sensor you can view at this link
:" & vbCrLf
        txt2      =      txt2      &
"://learn.adafruit.com/adafruit-optical-
fingerprint-sensor/wiring-for-use-with-
arduino"
        txt      =
"http://playground.arduino.cc/Main/DHT11Lib"
        'Case 4
        '          txt      =
"https://www.arduino.cc/en/Main/arduinoBoard
Due"
        'Case 5
        '          txt      =
"https://www.arduino.cc/en/Main/arduinoBoard
ProMini"
        'Case 6
        '          txt      =
"https://www.arduino.cc/en/main/arduinoBoard
Leonardo"
        'Case 7
        '          txt      =
"https://www.arduino.cc/en/Main/arduinoBoard
LilyPad"
        'Case Else
        ' txt = ""
    End Select
End If

    TXT_View.Text = txt2 & txt
End Sub
End Class

```

12. Buat form terahir untuk menampilkan program seperti gambar 3.21.



Gambar 3.21 Membuat *Form* Program Yang Akan Diunggah ke Arduino

13. Isikan kode berikut

```
Public Class Form1

    Private Sub AboutToolStripMenuItem_Click(sender As Object, e As EventArgs) Handles AboutToolStripMenuItem.Click
        MsgBox("Easy Arduino Programming V1" & vbCrLf & "Copyright @2017", MsgBoxStyle.Information, "About")
        AboutBox1.Show()
    End Sub

    Private Sub HelpToolStripMenuItem1_Click(sender As Object, e As EventArgs) Handles HelpToolStripMenuItem1.Click
        MsgBox("For more information, please contact us at 0823-xxxx-xxxx", MsgBoxStyle.Information, "Help")
    End Sub

    Private Sub UserControl11_Load(sender As Object, e As EventArgs)
    End Sub
```

```

        Private Sub ExitToolStripMenuItem_Click(sender As Object, e As EventArgs) Handles ExitToolStripMenuItem.Click
            Me.Close()
        End Sub

        Private Sub MainMenu1_Load(sender As Object, e As EventArgs) Handles MainMenu1.Load
        End Sub

        Private Sub MenuStrip1_ItemClicked(sender As Object, e As ToolStripItemClickedEventArgs) Handles MenuStrip1.ItemClicked
        End Sub
    End Class

    Public Class Finish

        Private Sub Finish_Load(sender As Object, e As EventArgs) Handles MyBase.Load
            note_txt.Text = "Setting is complete" & vbCrLf & "You can download the source code" & vbCrLf & "or go back to edit the settings"
        End Sub

        Private Sub Button1_Click(sender As Object, e As EventArgs) Handles Button1.Click
            Form1.SettingPin1.Visible = True
        End Sub

        Private Sub Button2_Click(sender As Object, e As EventArgs) Handles Button2.Click
    
```

```

        MsgBox(sensorSelected.ToString &
vbCrLf & arduinoSelected.ToString)

    End Sub
End Class

Finish code

```

Proses pembuatan jendela berikutnya sama halnya dengan jendela-jendela yang sebelumnya telah di buat akan di cantumkan lebih lanjut pada lampiran

3.4.2 Proses Penggabungan Pekerjaan Blok C

Setelah masing – masing area Blok dari gambar flowchart pembagian tugas kelompok, baik area blok “A” yang ditugaskan kepada mahasiswa pertama dan area blok “B” yang ditugaskan kepada mahasiswa kedua, maka semua tugas dari masing – masing mahasiswa akan digabungkan dalam area blok “C”. Dalam bagian area blok “B: ini, *hardware* yang telah didata serta disiapkan, sudah disediakan *database* program yang di butuhkan *prototype* oleh bagian anggota kelompok yang lain, sehingga *prototype box* pengaman perangkat BTS (*Base Transceiver Station*) dapat berjalan sesuai dengan uraian yang telah di jelaskan dalam bab sebelumnya.

Setelah program tersebut di tersebut diaplikasikan oleh anggota kelompok yang lain pada *prototype*, maka langkah selanjutnya yaitu membuat sebuah software sebagai *interface* menggunakan Visual Basic untuk menghubungkan bahasa pemrograman tersebut dengan *user* dengan membuat tampilan sederhana mungkin agar mudah dalam pengoperasiannya, serta tidak merubah fungsi dari program yang telah di siapkan oleh mahasiswa tersebut berdasarkan fungsi dari *prototype Box* Pengaman Perangkat Bts (*Base Transceiver Station*) itu sendiri.

Setelah software untuk *interface* telah selesai, maka langkah selanjutnya yaitu menuju penggabungan masing – masing tugas dari kedua anggota kelompok. Proses penggabungan dimulai dengan memasukkan *database* program tersebut kedalam *interface* yang sebelumnya telah dibuat menggunakan Visual Basic sebagai penghubung kepada *user* seperti yang telah dijelaskan langkah – langkahnya pada sub bab

sebelumnya. Setelah itu program valid untuk *prototype Box Pengaman Perangkat Bts (Base Transceiver Station)* yang telah saya masukan pada *interface* yang sebelumnya telah dibuat sebelumnya sudah dapat di katakan *Generic Programmable Penghubung Pengguna Dengan I/O Dari Arduino Yang Diterapkan Pada Box Pengaman Perangkat Bts (Base Transceiver Station)* yang kami beri nama “*Do It By Yourself*” di mana software ini dan dapat memudahkan user yang notabennya adalah orang awam dapat dengan mudah membuat sebuah rangkaian elektronika berbasis arduino tanpa adanya kendala dengan bahasa pemrograman arduino yang cukup rumit.

Selanjutnya proses validasi akan di bahas lebih lanjut di bab 4 dimana proses ini ialah proses pengambilan data saat kedua alur yaitu alur hijau dan biru sudah di gabungkan menjadi satu.

-----Halaman ini sengaja dikosongkan-----

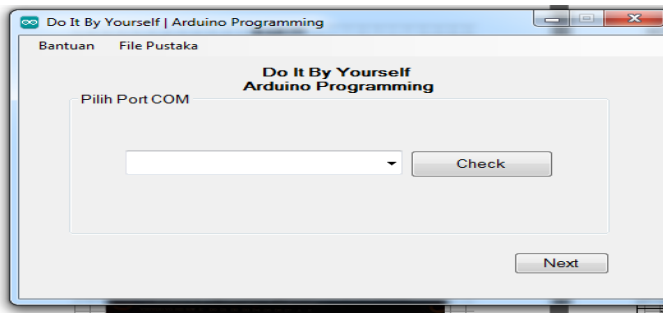
BAB IV

PENGUJIAN DAN ANALISA DATA

Untuk mengetahui bahwa alat telah bekerja dengan benar maka perlu dilakukan pengujian alat yang meliputi pengujian perangkat keras dan pengujian perangkat lunak. Pengujian yang dilakukan pada peralatan untuk mengetahui kesesuaian antara teori dengan hasil perancangan, yaitu dengan mengetahui hasil pengukuran pada setiap perangkat yang telah dibuat.

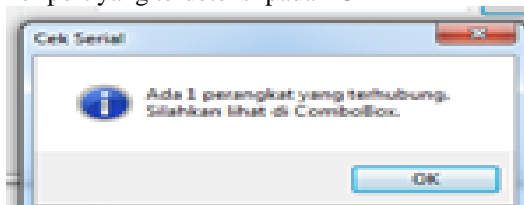
4.1 Pengujian Software

Jendela pertama di gunakan untuk mendeteksi dan memilih port arduino pada PC agar dapat memulai proses pemrograman. Pada gambar 4.1 dibawah ini pada saat awal sebelum arduino belum di hubungkan ke PC



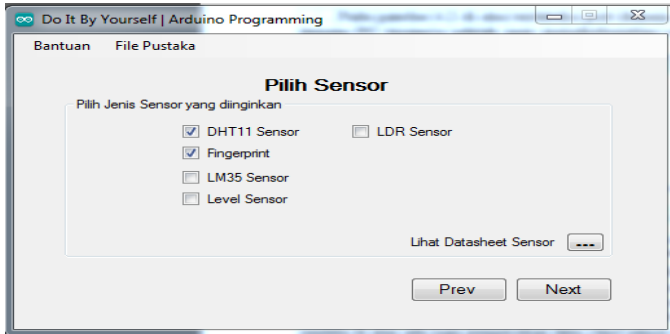
Gambar 4.1 Tampilan Jendela Pemilihan Port

Pada gambar 4.2 di bawah ini arduino di deteksi secara otomatis dengan nomer port yang terdeteksi pada PC



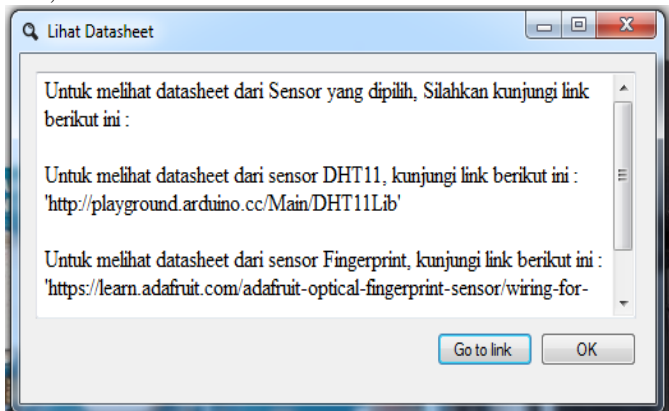
Gambar 4.2 Tampilan Jendela Saat Port Arduino Terdeteksi

Pada gambar 4.2 di atas terdeteksi port dimana arduino terhubung dengan PC tujuannya adalah agar menghubungkan *software* “Do It By Your Self” dengan arduino.



Gambar 4.3 Tampilan Jendela Pemilihan Sensor Yang Akan Dipakai

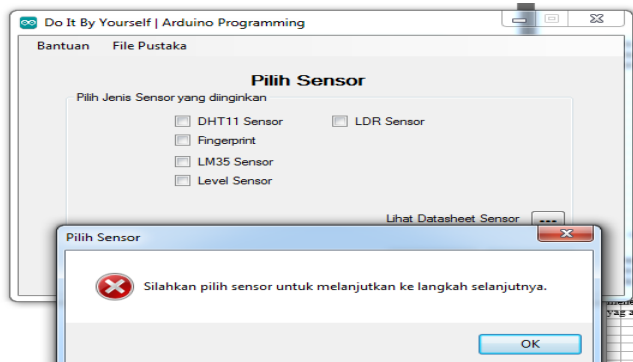
Pada gambar 4.3 di atas *user* di hadapkan oleh jenis sensor *input* yang akan di gunakan. Dalam *software* ini hanya di batasi hanya menggunakan DHT dan *fingerprint* sensor karena kebutuhan pengembangan dan penerapanya hanya menggunakan jenis sensor itu. Selain itu pada jendela di atas ada juga pengecekan data *sheet* sensor *input* yang telah di pilih maka pada saat user menekan menu tersebut akan langsung otomatis diarahkan ke web yang menyediakan data *sheet* ari sensor *input* yang telah di pilih. Contohnya pada gambar 4.4 di bawah ini,



Gambar 4.4 Tampilan Jendela *Link Data Sheet* Sensor Yang Dipilih

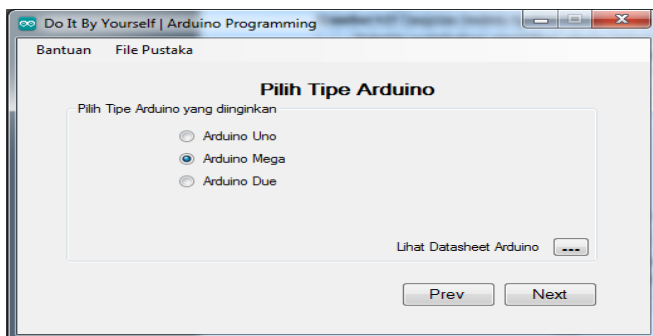
Pada gambar 4.4 di atas di tampilkan *link* dari sensor yang sebelumnya telah di pilih pada gambar 4.3.

Namun apabila kita tidak memilih sensor yang akan di pakai maka akan tampil pemberitahuan bahwa kita harus memilih sensor yang akan di gunakan. Dalam hal ini yang kita khususkan yang akan di terapkan pada pengembangannya. Contoh pemeberituannya dapat di lihat pada gambar 4.5 di bawah ini



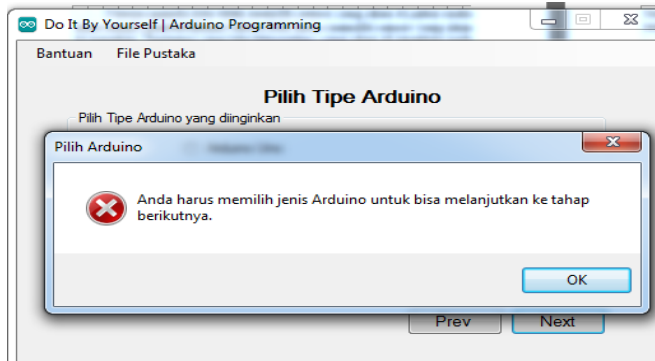
Gambar 4.5 Tampilan Jendela Apabila Sensor Belom Dipilih

Setelah melakukan pemilihan sensor *input* yang akan di gunakan maka pada saat user menekan tombol *next* maka akan keluar jendela pemilihan tipe arduino yang akan di gunakan, seperti gambar 4.6 di bawah ini.



Gambar 4.6 Tampilan Jendela Pemilihan Tipe Arduino

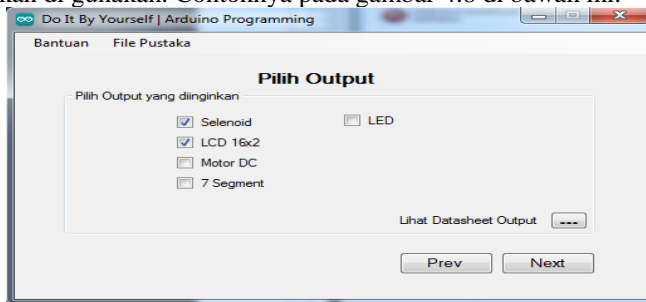
Di dalam gambar 4.6 diatas juga tetraera kolom untuk melihat data sheet yang akan di gunakan, apabila kita sudah memilih tipe arduino yang sudah di gunakan dan menekan kolom tersebut akan di arahkan ke pemilihan *output*. Selanjutnya sama halnya jendela pemilihan sensor apabila kita tidak memilih tipe arduino maka akan muncul jendela peringatan seperti gambar 4.7.



Gambar 4.7 Jendela Peringatan Tidak Memilh Tipe Arduino

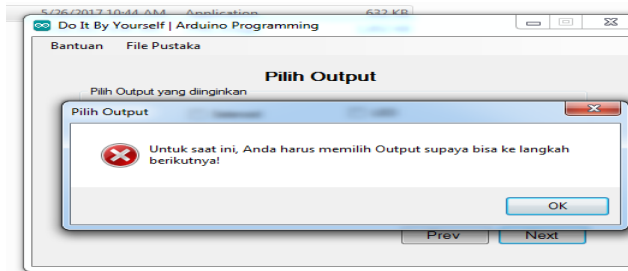
Dalam hal ini kita hanya menyesuaikan arduino dengan hanya mengharuskan 3 tipe arduino. Khususnya dalam hal ini menggunakan Arduino tipe mega 2560 Karena menyesuaikan denga *prototype*.. Sebenarnya tipe arduino yang di gunakan dapat di atur dengan mudah, apabila memenuhi standart sensor yang di gunakan.

Selanjutnya setelah memilih tipe arduino yang di gunakan dan menekan tombol *next* maka akan tampil jendela pemilihan jenis output yag akan di gunakan. Contohnya pada gambar 4.8 di bawah ini.



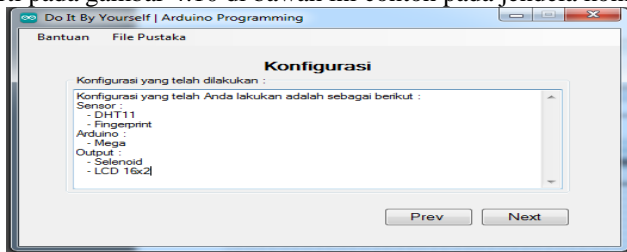
Gambar 4.8 Tampilan Jendela Pemilihan Jenis Sensor *Output*

Pada saat kondisi *user* tidak memilih jenis sensor *output* yang akan di gunakan maka proses tidak dapat berjalan dan akan muncul jendela pemberitahuan jenis sensor yang akan di gunakan, dalam hal ini jenis sensor *output* yang akan di gunakan di sesuai kan dengan *prototype*, yaitu selenoid,LCD16x4 karena pada *prototype* dibutuhkan *output* jenis ini guna menampilkan status sensor DHT11 dan selenoid sebagai *output*an fingerprint Seperti contoh gambar 4.9 di bawah ini



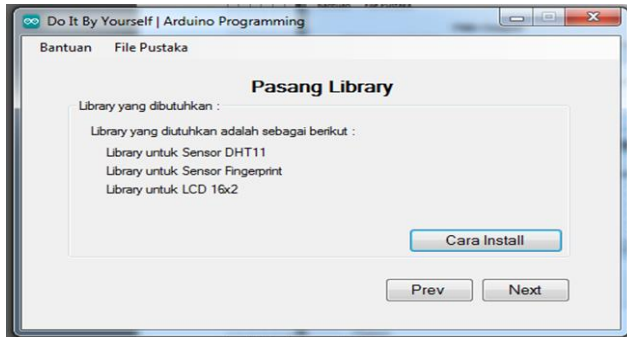
Gambar 4.9 Jendela Peringatan Tidak Memilih Jenis *Output*

Selain itu pada gambar 4.8 juga terdapat tombol untuk melihat data sheet sensor yang telah di pilih sebelumnya. Pada saat kita sudah menentukan jenis *output* yang akan di pakai, maka proses akan berlanjut setelah kita menekan tombol *next*. Setelah kita menekan tombol *next* maka akan muncul jendela konfigurasi, dimana jendela ini berisi tentang informasi jenis *input* dan *output* yang telah kita pilih sebelumnya dan juga jenis arduino yang telah kita pilih sebelumnya, tujuannya ialah memberitahu *user* tentang konfigurasi yang telah di lakukan, apabila ada sebuah kesalahan maka bisa di koreksi dengan cara menekan tombol *prev* sampai pada bagian jendela konfigurasi yang di rasa kurang tepat. Seperti pada gambar 4.10 di bawah ini contoh pada jendela konfigurasi



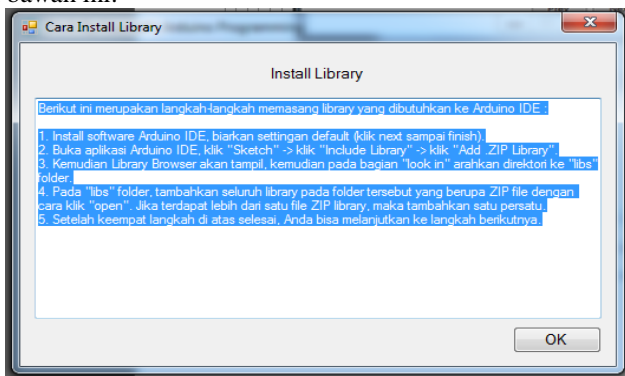
Gambar 4.10 Tampilan Jendela Konfigurasi

Apabila dirasa konfigurasi yang telah di atur sudah benar maka proses dapat di lanjutkan dengan menekan tombol *next*. Setelah kita menekan tombol *next* maka akan keluar jendela pemasangan *library* sensor *input* maupun *output* yang telah di pilih sebelumnya. Contohnya seperti gambar 4.11 di bawah ini



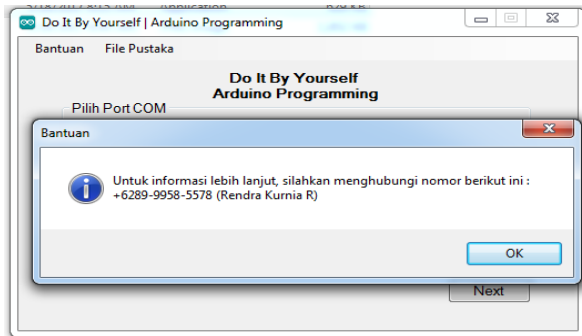
Gambar 4.11 Tampilan Jendela *Library* Yang Dibutuhkan

Setelah itu cara menginstal *library* sensor dapat kita lihat dengan menekan tombol cara instal dan akan muncul jendela baru yang menunjukan kepada *user* bagaimana cara menginstal *library* dari snesor yang telah kita pilih sebelumnya. *User* hanya perlu mengikuti petunjuk pada jendela baru yang tampil tersebut. Ikuti langkah-langkahnya perlahan-lahan sesuai dengan petunjuknya. Contohnya seperti gambar 4.12 di bawah ini.



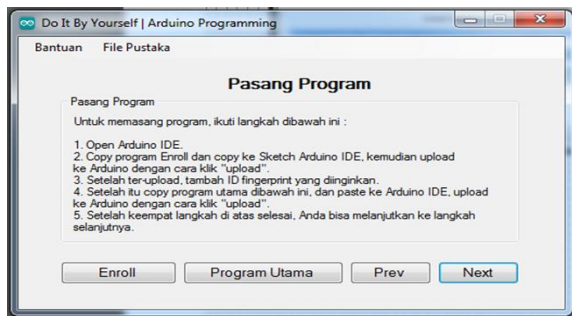
Gambar 4.12 Tampilan Jendela Petunjuk Cara Menginstal *Library*

Apabila mengalami kesulitan dalam menggunakan program ini telah di sediakan pula kolom bantuan yang dapat di pilih pada kolom utama dan setelah di tekan maka akan muncul *contact* bantuan yang dapat di hubungi seperti gambar 4.13 di bawah ini.



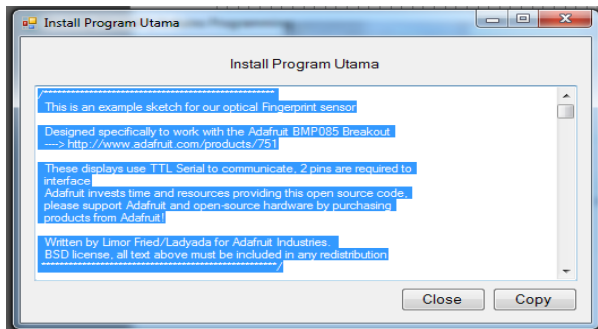
Gambar 4.13 Tampilan Jendela Bantuan & *Contact Person*

Setelah proses pemasangan *library* maka saat pengguna melanjutkan ke proses berikutnya dengan memilih tombol *next* maka akan tampil jendela pemasangan program. Pada jendela pemasanga program akan di tampilkan bagaimana cara mengupload program yang telah di susun oleh software sesuai dengan kebutuhan *user* yang telah di konfigurasi sebelumnya. Selain itu di sana juga terdapat langkah-langkah yang harus di lalui *user* agar program yang telah di susun oleh software ini dapat di jalankan dengan sempurna contohnya pada gambar 4.14 di bawah ini menu jendela pemasangan program.



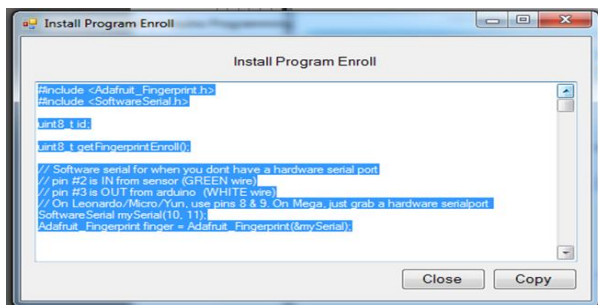
Gambar 4.14 Tampilan Jendela Cara Pemasangan Program

Setelah itu pada saat *user* memilih kolom program utama maka akan tampil bahasa pemrograman arduino sesuai dengan konfigurasi *input* dan *output* yang telah di atur sebelumnya seperti pada gambar 4.15 di bawah ini



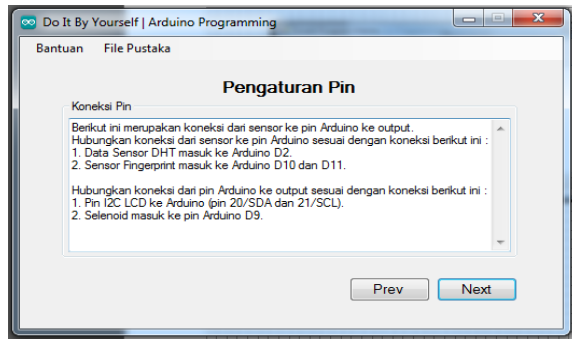
Gambar 4.15 Tampilan Jendela Program Utama

Sedangkan pada jendela pemasangan program terdapat kolom bertuliskan *enroll* ini berisi tentang program yang di kususkan dengan *prototype* yang di buat menggunakan sensor *fingerprint*. Dimana tujuan dari *enroll* ini sendiri merupakan program yang di gunakan untuk memasukan data id *finger* dari pengguna yang nantinya akan di simpan pada memori dari sensor itu sendiri dan dapat di baca setelah sistem yang telah di rancang tersebut berjalan. Selain itu pada saat kita menggunakan sensor *fingerprint* program *enroll* harus *upload* dahulu sebelum program utama *upload* seperti contoh gambar 4.16 di bawah ini.



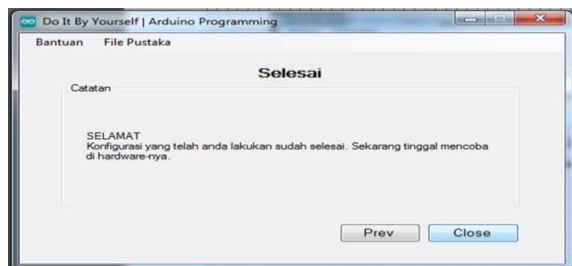
Gambar 4.16 Tampilan Jendela Program *Enroll*

Namun *user* tidak perlu mengupload program *enroll* apabila pada konfigurasi yang di pilih tidak menggunakan sensor *fingerprint*. Setelah program arduino pada gambar 4.14 di lakukan maka proses selanjutnya apabila memilih kolom *next* akan tampil jendela pengaturan pin *input* dan *output* yang sebelumnya telah di konfigurasi oleh pengguna. Seperti gambar 4.17 di bawah ini



Gambar 4.17 Tampilan Jendela Pengaturan Pin

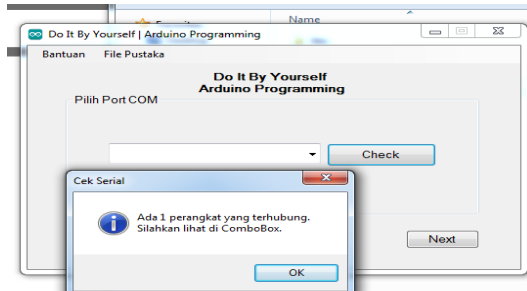
Pada jendela pengaturan pin ini *user* tinggal mengikuti arahan dan cara menghubungkan pin *input* sensor yang menuju ke arduino maupun output arduino yang menuju ke sensor, beserta pengalamatannya. Setelah proses pada gambar 4.17 di lalui maka pada saat *user* memilih kolom *next* proses akan berakhir dengan jendela pemberitahuan bahwa proses telah usai, dan hardware yang telah di isi dengan bahasa pemrograman secara otomatis menggunakan *software* “*Do It By Yourself*” ini telah siap di coba. Seperti contoh pada gambar 4.18 di bawah ini.



Gambar 4.18 Tampilan Jendela Pemberitahuan Proses Telah Selesai

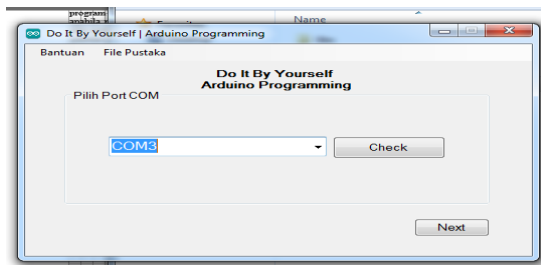
4.2 Pengujian Mengubungkan Software

Pada bab ini akan di uji cobakan untutk penerapan *software* untuk memprogram Arduino pertama-tama pengujian di awali dengan pendeteksi port pada *software*. Apabila kita menekan tombol *check* apabila terdeteksi maka akan muncul pemberitahuan seperti pada gambar 4.19 di bawah ini



Gambar 4.19 Tampilan Jendela Pemberitahuan Port Terdeteksi

Pada gambar di atas terdapat pemberitahuan bahwa terdapat 1 perangkat yang terhubung artinya komunikasi dengan Arduino telah berhasil dan proses dalam menentukan *input output* akan dapat berjalan. Karena proses ini sangat penting dalam menghubungkan Arduino karena komunikasi antara Arduino dan komputer dapat dilakukan melalui port serial (via USB). Dalam hal ini, Arduino tidak hanya bisa membaca data dari komputer yang ada di port serial, melainkan juga dapat mengirim data ke komputer. Sehingga komunikasi yang dilakukan bersifat dua arah. Apabila telah muncul pemberitahuan seperti pada gambar 4.19 di atas maka kita hanya perlu memilih port yang telah di deteksi di *combobox* seperti pada gambar 4.20 di bawah ini

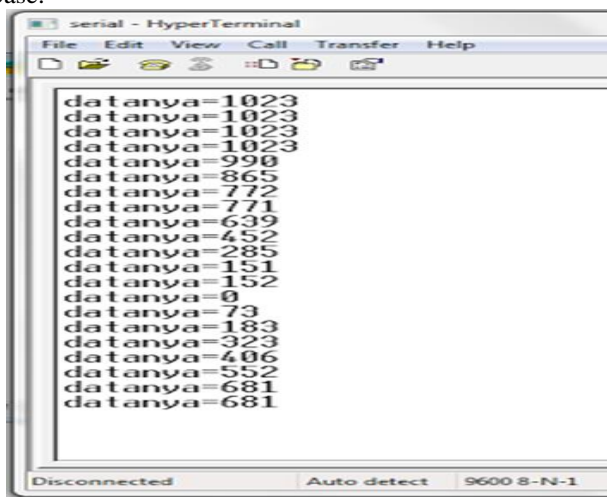


Gambar 4.20 Tampilan Jendela Pemilihan Port Berhasil

Setelah jendela di atas tampil tandanya arduino telah terhubung dengan *software* dengan demikian proses selanjutnya hanya dengan mengatur keluaranya saja seperti yang telah di jelaskan pada bab 4.1

Dari sourcecode di bab 3.4.1 dapat kita lihat bahwa tidak ada coding yang aneh, Arduino membaca nilai sensor kemudian dikirim ke serial dengan perintah `Serial.print(cm)`. Nilai tersebut akan diterima di komputer untuk selanjutnya dimasukan ke dalam database. Untuk mengecek apakah data dari serial tersebut dapat diterima oleh komputer, kita bisa melihatnya pada serial monitor yang terdapat pada Arduino. Seperti gambar 4.21 lalu, bagaimana proses masuk ke dalam databasenya?

Data yang masuk via serial agar bisa masuk ke dalam database saya lakukan dengan cara membuat program menggunakan Visual Basic,. Ilmu ini saya dapatkan dari artikel terkait Arduino dan Database kebanyakan arduino terebut terhubung ke web server dan database via Wifi atau Ethernet, maka dari itu saya di sini ingin menyajikan dengan cara berbeda yakni Sensor -> Arduino -> Serial Ke komputer -> Diterima oleh aplikasi yang dibuat pakai Visual Basic -> simpan ke database.



Gambar 4.21 Tampilan Jendela Pemilihan Port Berhasil

-----Halaman ini sengaja dikosongkan-----

BAB V

PENUTUP

Setelah melakukan perencanaan, perancangan, dan pengujian alat maka dapat diambil kesimpulan dan memberikan saran demi penyempurnaan Tugas Akhir ini.

5.1 Kesimpulan

Hasil dari pengujian dari masing – masing database program yang di pakai pada rancangan alat, maka dapat disimpulkan bahwa :

1. Langkah – langkah yang harus di lalui untuk dapat menggunakan software harus sesuai dengan petunjuk agar dapat berjalan sesuai dengan yang di harapkan
2. Keberhasilan komunikasi serial yang di buat otomatis dengan *software* menentukan proses kerja dari software
3. Kemungkinan *bug software* sudah tidak mungkin terjadi karena kemungkinan itu telah di minimalisir

5.2 Saran dan Masukan

Untuk pengembangan *software* ini selanjutnya di sediakan sensor berupa *input ouput* yang beragam agar nantinya *software* ini dapat menjadi *software* yang benar-benar *generic* dan juga dapat berguna untuk segala kondisi yang memungkinkan. Karena pada *software* ini ada masalah yang timbul berkaitan dengan keterbatasan biaya yang membuat sensor *input,output*, yang di *generickan* hanya terbatas.

-----Halaman ini sengaja dikosongkan-----

DAFTAR PUSTAKA

- [1] Arief, H. S., Perbandingan Akurasi Pengukuran Suhu dan Kelembaban Antara Sensor DHT11 dan SHT11, Jurnal Infotel Vol. 6 No. 2, Sekolah Tinggi Teknologi Telematika Telkom Purwokerto, 2014.
- [2] Kadir, A., Panduan Praktis Mempelajari Aplikasi Mikrokontroler dan Pemrogramannya Menggunakan Arduino, Edisi 1 ISBN 978-979-29-4017-6, ANDI Publisher, Yogyakarta, 2013.
- [3] Kaur, M, Singh, M, Girdar, A and S.Sandhu, P. Fingerprint Verification System using Minutiae Extraction Technique. *World Academy of Science, Engineering and Technology* 46. pp. 497-502., 2008
- [4] Munir, R., Pengenalan Microsoft Visual Basic 6.0 dan Interface. Bandung: Infomatika, Bandung, 2004.
- [5] Kurniadi, Adi, Pemrograman Microsoft Visual Basic, Penerbit PT. Elek Media, Yogyakarta, 2003.
- [6] Spring Loaded Elektromagnet. Solenoid 12V Pull Type. [Online]. Tersedia : <http://www.engineeringshock.com/12v-pull-type-solenoid.html>
- [7] Sunyoto, Andi, Pemrograman Database dengan Visual Basic dan Microsoft SQL, Penerbit ANDI, Yogyakarta, 2007.
- [8] <http://ecadio.com/belajar-dan-mengenal-sensor-input-output>
- [9] <http://www.ngarep.net/input-output-analog-dan-digital/>

-----Halaman ini sengaja dikosongkan-----

LAMPIRAN A

A.1. Source Program Visual Basic

*program jendela awal

```
<?xml version="1.0" encoding="utf-8"?>
<Project ToolsVersion="4.0"
DefaultTargets="Build"
xmlns="http://schemas.microsoft.com/developer/msbuild/
2003">
  <Import
Project="$(MSBuildExtensionsPath)\$(MSBuildToolsVersio
n)\Microsoft.Common.props"
Condition="Exists('$(MSBuildExtensionsPath)\$(MSBuildT
oolsVersion)\Microsoft.Common.props') " />
  <PropertyGroup>
    <Configuration Condition=" '$(Configuration)'
== ' ' ">Debug</Configuration>
    <Platform Condition=" '$(Platform)' == ' '
">AnyCPU</Platform>
    <ProjectGuid>{6F177C79-A643-4DE4-8C20-
D0AF064469B8}</ProjectGuid>
    <OutputType>WinExe</OutputType>

    <StartupObject>WindowsApplication1.My.MyApplication</S
tartupObject>

    <RootNamespace>WindowsApplication1</RootNamespace>
    <AssemblyName>Arduino
Programming</AssemblyName>
    <FileAlignment>512</FileAlignment>
    <MyType>WindowsForms</MyType>

    <TargetFrameworkVersion>v4.5</TargetFrameworkVersion>
  </PropertyGroup>
  <PropertyGroup Condition="
'$(Configuration)|$(Platform)' == 'Debug|AnyCPU' ">
    <PlatformTarget>AnyCPU</PlatformTarget>
    <DebugSymbols>true</DebugSymbols>
    <DebugType>full</DebugType>
    <DefineDebug>true</DefineDebug>
    <DefineTrace>true</DefineTrace>
    <OutputPath>bin\Debug\</OutputPath>
```

```

        <DocumentationFile>Arduino
Programming.xml</DocumentationFile>

<NoWarn>42016,41999,42017,42018,42019,42032,42036,4202
0,42021,42022</NoWarn>
    </PropertyGroup>
    <PropertyGroup Condition="
'$ (Configuration) |$(Platform)' == 'Release|AnyCPU' ">
        <PlatformTarget>AnyCPU</PlatformTarget>
        <DebugType>pdbonly</DebugType>
        <DefineDebug>>false</DefineDebug>
        <DefineTrace>>true</DefineTrace>
        <Optimize>>true</Optimize>
        <OutputPath>bin\Release\</OutputPath>
        <DocumentationFile>Arduino
Programming.xml</DocumentationFile>

<NoWarn>42016,41999,42017,42018,42019,42032,42036,4202
0,42021,42022</NoWarn>
    </PropertyGroup>
    <PropertyGroup>
        <OptionExplicit>On</OptionExplicit>
    </PropertyGroup>
    <PropertyGroup>
        <OptionCompare>Binary</OptionCompare>
    </PropertyGroup>
    <PropertyGroup>
        <OptionStrict>Off</OptionStrict>
    </PropertyGroup>
    <PropertyGroup>
        <OptionInfer>On</OptionInfer>
    </PropertyGroup>
    <PropertyGroup>

<ApplicationIcon>arduino_128px_1171938_easyicon.net.ic
o</ApplicationIcon>
    </PropertyGroup>
    <ItemGroup>
        <Reference Include="System" />
        <Reference Include="System.Activities" />
        <Reference Include="System.Data" />
        <Reference Include="System.Deployment" />
        <Reference Include="System.Drawing" />
        <Reference Include="System.ServiceModel" />

```

```

        <Reference
Include="System.ServiceModel.Activities" />
        <Reference Include="System.Windows.Forms" />
        <Reference Include="System.Xaml" />
        <Reference Include="System.Xml" />
        <Reference Include="System.Core" />
        <Reference Include="System.Xml.Linq" />
        <Reference
Include="System.Data.DataSetExtensions" />
    </ItemGroup>
    <ItemGroup>
        <Import Include="Microsoft.VisualBasic" />
        <Import Include="System" />
        <Import Include="System.Collections" />
        <Import Include="System.Collections.Generic"
/>
        <Import Include="System.Data" />
        <Import Include="System.Drawing" />
        <Import Include="System.Diagnostics" />
        <Import Include="System.Windows.Forms" />
        <Import Include="System.Linq" />
        <Import Include="System.Xml.Linq" />
        <Import Include="System.Threading.Tasks" />
    </ItemGroup>
    <ItemGroup>
        <Compile Include="AboutBox1.Designer.vb">
            <DependentUpon>AboutBox1.vb</DependentUpon>
        </Compile>
        <Compile Include="AboutBox1.vb">
            <SubType>Form</SubType>
        </Compile>
        <Compile Include="Cara_Install.Designer.vb">
            <DependentUpon>Cara_Install.vb</DependentUpon>
        </Compile>
        <Compile Include="Cara_Install.vb">
            <SubType>Form</SubType>
        </Compile>
        <Compile
Include="Configuration_window.Designer.vb">
            <DependentUpon>Configuration_window.vb</DependentUpon>
        </Compile>
        <Compile Include="Configuration_window.vb">

```

```

        <SubType>UserControl</SubType>
    </Compile>
    <Compile Include="Finish.Designer.vb">
        <DependentUpon>Finish.vb</DependentUpon>
    </Compile>
    <Compile Include="Finish.vb">
        <SubType>UserControl</SubType>
    </Compile>
    <Compile Include="Form1.vb">
        <SubType>Form</SubType>
    </Compile>
    <Compile Include="Form1.Designer.vb">
        <DependentUpon>Form1.vb</DependentUpon>
        <SubType>Form</SubType>
    </Compile>
    <Compile
Include="Install_Enroll.Designer.vb">

    <DependentUpon>Install_Enroll.vb</DependentUpon>
    </Compile>
    <Compile Include="Install_Enroll.vb">
        <SubType>Form</SubType>
    </Compile>
    <Compile
Include="Install_Library_Window.Designer.vb">

    <DependentUpon>Install_Library_Window.vb</DependentUpon>
n>
    </Compile>
    <Compile Include="Install_Library_Window.vb">
        <SubType>UserControl</SubType>
    </Compile>
    <Compile
Include="Install_Program.Designer.vb">

    <DependentUpon>Install_Program.vb</DependentUpon>
    </Compile>
    <Compile Include="Install_Program.vb">
        <SubType>Form</SubType>
    </Compile>
    <Compile
Include="Install_Program_Window.Designer.vb">

```

```

<DependentUpon>Install_Program_Window.vb</DependentUpon>
n>
    </Compile>
    <Compile Include="Install_Program_Window.vb">
        <SubType>UserControl</SubType>
    </Compile>
    <Compile Include="Module1.vb" />
    <Compile Include="My Project\AssemblyInfo.vb"
/>
    <Compile Include="My
Project\Application.Designer.vb">
        <AutoGen>True</AutoGen>

<DependentUpon>Application.myapp</DependentUpon>
    </Compile>
    <Compile Include="My
Project\Resources.Designer.vb">
        <AutoGen>True</AutoGen>
        <DesignTime>True</DesignTime>

<DependentUpon>Resources.resx</DependentUpon>
    </Compile>
    <Compile Include="My
Project\Settings.Designer.vb">
        <AutoGen>True</AutoGen>

<DependentUpon>Settings.settings</DependentUpon>

<DesignTimeSharedInput>True</DesignTimeSharedInput>
    </Compile>
    <Compile Include="MainMenu.Designer.vb">
        <DependentUpon>MainMenu.vb</DependentUpon>
    </Compile>
    <Compile Include="MainMenu.vb">
        <SubType>UserControl</SubType>
    </Compile>
    <Compile Include="SelectArduino.Designer.vb">

<DependentUpon>SelectArduino.vb</DependentUpon>
    </Compile>
    <Compile Include="SelectArduino.vb">
        <SubType>UserControl</SubType>
    </Compile>

```

```

        <Compile Include="SelectSensor.Designer.vb">

<DependentUpon>SelectSensor.vb</DependentUpon>
    </Compile>
    <Compile Include="SelectSensor.vb">
        <SubType>UserControl</SubType>
    </Compile>
    <Compile Include="Select_Output.Designer.vb">

<DependentUpon>Select_Output.vb</DependentUpon>
    </Compile>
    <Compile Include="Select_Output.vb">
        <SubType>UserControl</SubType>
    </Compile>
    <Compile Include="SettingPin.Designer.vb">

<DependentUpon>SettingPin.vb</DependentUpon>
    </Compile>
    <Compile Include="SettingPin.vb">
        <SubType>UserControl</SubType>
    </Compile>
    <Compile
Include="view_datasheet.Designer.vb">

<DependentUpon>view_datasheet.vb</DependentUpon>
    </Compile>
    <Compile Include="view_datasheet.vb">
        <SubType>Form</SubType>
    </Compile>
</ItemGroup>
<ItemGroup>
    <EmbeddedResource Include="AboutBox1.resx">
        <DependentUpon>AboutBox1.vb</DependentUpon>
    </EmbeddedResource>
    <EmbeddedResource
Include="Cara_Install.resx">

<DependentUpon>Cara_Install.vb</DependentUpon>
    </EmbeddedResource>
    <EmbeddedResource
Include="Configuration_window.resx">

<DependentUpon>Configuration_window.vb</DependentUpon>
    </EmbeddedResource>

```

```

        <EmbeddedResource Include="Finish.resx">
            <DependentUpon>Finish.vb</DependentUpon>
        </EmbeddedResource>
        <EmbeddedResource Include="Form1.resx">
            <DependentUpon>Form1.vb</DependentUpon>
        </EmbeddedResource>
        <EmbeddedResource
Include="Install_Enroll.resx">

        <DependentUpon>Install_Enroll.vb</DependentUpon>
        </EmbeddedResource>
        <EmbeddedResource
Include="Install_Library_Window.resx">

        <DependentUpon>Install_Library_Window.vb</DependentUpo
n>
        </EmbeddedResource>
        <EmbeddedResource
Include="Install_Program.resx">

        <DependentUpon>Install_Program.vb</DependentUpon>
        </EmbeddedResource>
        <EmbeddedResource
Include="Install_Program_Window.resx">

        <DependentUpon>Install_Program_Window.vb</DependentUpo
n>
        </EmbeddedResource>
        <EmbeddedResource                                Include="My
Project\Resources.resx">

        <Generator>VbMyResourcesResXFileCodeGenerator</Generat
or>

        <LastGenOutput>Resources.Designer.vb</LastGenOutput>

        <CustomToolNamespace>My.Resources</CustomToolNamespace
>
            <SubType>Designer</SubType>
        </EmbeddedResource>
        <EmbeddedResource Include="MainMenu.resx">
            <DependentUpon>MainMenu.vb</DependentUpon>
        </EmbeddedResource>

```

```

        <EmbeddedResource
Include="SelectArduino.resx">

<DependentUpon>SelectArduino.vb</DependentUpon>
        </EmbeddedResource>
        <EmbeddedResource
Include="SelectSensor.resx">

<DependentUpon>SelectSensor.vb</DependentUpon>
        </EmbeddedResource>
        <EmbeddedResource
Include="Select_Output.resx">

<DependentUpon>Select_Output.vb</DependentUpon>
        </EmbeddedResource>
        <EmbeddedResource Include="SettingPin.resx">

<DependentUpon>SettingPin.vb</DependentUpon>
        </EmbeddedResource>
        <EmbeddedResource
Include="view_datasheet.resx">

<DependentUpon>view_datasheet.vb</DependentUpon>
        </EmbeddedResource>
    </ItemGroup>
    <ItemGroup>
        <None Include="My Project\Application.myapp">

<Generator>MyApplicationCodeGenerator</Generator>

<LastGenOutput>Application.Designer.vb</LastGenOutput>
        </None>
        <None Include="My Project\Settings.settings">

<Generator>SettingsSingleFileGenerator</Generator>

<CustomToolNamespace>My</CustomToolNamespace>

<LastGenOutput>Settings.Designer.vb</LastGenOutput>
        </None>
        <None Include="App.config" />
    </ItemGroup>
    <ItemGroup>

```



```

        <Content
Include="arduino_128px_1171938_easyicon.net.ico" />
        </ItemGroup>
        <Import
Project="$(MSBuildToolsPath)\Microsoft.VisualBasic.targets" />
        <!-- To modify your build process, add your
task inside one of the targets below and uncomment it.
        Other similar extension points exist, see
Microsoft.Common.targets.
        <Target Name="BeforeBuild">
        </Target>
        <Target Name="AfterBuild">
        </Target>
        -->
    </Project>

```

***Program pilih sensor**

```
Public Class SelectSensor
```

```

    Private Sub Button1_Click(sender As Object, e
As EventArgs) Handles Button1.Click
        Form1.MainMenu1.Visible = True
    End Sub

```

```

    Private Sub Button2_Click(sender As Object, e
As EventArgs) Handles Button2.Click
        selectSensor()
        If sensorSelected = 0 Then
            myMsg()
        Else
            Me.Visible = False
        End If
    End Sub

```

```

    Private Sub Button3_Click(sender As Object, e
As EventArgs) Handles PB_ViewDBSensor.Click
        sensorSelected = 0
        selectSensor()
        menuSelected = 1
        If sensorSelected = 0 Then
            myMsg2()
        Else
            'MsgBox(sensorSelected.ToString)

```

```

        view_datasheet.Show()

        'Select Case sensorSelected
        '    Case 1
        '
myMsg2("http://playground.arduino.cc/Main/DHT11Lib")
        '    Case 2
        '
myMsg2("https://learn.adafruit.com/adafruit-optical-
fingerprint-sensor/wiring-for-use-with-arduino")
        '    Case 3
        '
myMsg2("www.ti.com/lit/ds/symlink/lm35.pdf")
        '    Case 4
        '
        myMsg2("https://cdn-
shop.adafruit.com/datasheets/eTape+Datasheet+12110215T
C-12_040213.pdf")
        '    Case 5
        '
myMsg2("https://cdn.sparkfun.com/datasheets/Sensors/Pr
essure/Sensirion_Humidity_SHT1x_Datasheet_V5.pdf")
        '    Case 6
        '
myMsg2("https://www.sparkfun.com/products/10245")
        '    Case 7
        '
myMsg2("kennarar.vma.is/thor/v2011/vgr402/ldr.pdf")
        '    Case Else
        'End Select
    End If
End Sub

Private Sub SelectSensor_Load(sender As
Object, e As EventArgs) Handles MyBase.Load
    sensorSelected = 0
    menuSelected = 0

    RB_LM35.Enabled = True
    RB_LevelSensor.Enabled = True
    RB_LDR.Enabled = True

    'Inisialisasi Sensor
    dht11 = 0
    fingerprint = 0

```

```

        lm35 = 0
        level = 0
        ldr = 0

    End Sub

    Private Sub myMsg2()
        MsgBox("Anda belum memilih jenis Sensor!
        Silahkan pilih sensor dahulu.", MsgBoxStyle.Critical,
        "Pilih Sensor")
    End Sub

    Private Sub myMsg()
        MsgBox("Silahkan pilih sensor untuk
        melanjutkan ke langkah selanjutnya.",
        MsgBoxStyle.Critical, "Pilih Sensor")
    End Sub

    Private Sub selectSensor()
        'Get Selected Sensor
        If RB_DHT11.Checked Then
            sensorSelected = 1
            dht11 = 1
        Else
            dht11 = 0
        End If

        If RB_Fingerprint.Checked Then
            sensorSelected = 2
            fingerprint = 1
        Else
            fingerprint = 0
        End If

        If RB_LM35.Checked Then
            sensorSelected = 3
            lm35 = 1
        Else
            lm35 = 0
        End If

        If RB_LevelSensor.Checked Then
            sensorSelected = 4
            level = 1
        End If
    End Sub

```

```

Else
    level = 0
End If

If RB_LDR.Checked Then
    sensorSelected = 5
    ldr = 1
Else
    ldr = 0
End If

If Not (RB_DHT11.Checked Or
RB_Fingerprint.Checked Or RB_LM35.Checked Or
RB_LevelSensor.Checked Or RB_LDR.Checked) Then
    sensorSelected = 0
End If

'MsgBox(sensorSelected.ToString)
'MsgBox(dht11 & ", " & fingerprint & ", "
& lm35 & ", " & level & ", " & ldr,
MsgBoxStyle.Information, "")
End Sub
End Class

```

***Program pilih tipe Arduino**

```
Public Class SelectArduino
```

```

    Private Sub Button1_Click(sender As Object, e
As EventArgs) Handles Button1.Click
        Form1.SelectSensor1.Visible = True
    End Sub

```

```

    Private Sub Button2_Click(sender As Object, e
As EventArgs) Handles Button2.Click
        selectArduino()
        If arduinoSelected = 0 Then
            myMsg2()
        Else
            Me.Visible = False
            'If sensorSelected = 1 And
arduinoSelected = 1 Then
            'End If
        End If
    End Sub

```

```

        'MsgBox(arduinoSelected,
MsgBoxStyle.Information, "")
    End Sub

    Private Sub Button3_Click(sender As Object, e
As EventArgs) Handles PB_ViewDBArduino.Click
        arduinoSelected = 0
        selectArduino()
        'Flag for Select Sensor
        menuSelected = 0
        If arduinoSelected = 0 Then
            myMsg()
        Else
            view_datasheet.Show()
        End If
    End Sub

    Private Sub SelectArduino_Load(sender As
Object, e As EventArgs) Handles MyBase.Load
        arduinoSelected = 0
    End Sub

    Private Sub myMsg2()
        MsgBox("Anda harus memilih jenis Arduino
untuk bisa melanjutkan ke tahap berikutnya.",
MsgBoxStyle.Critical, "Pilih Arduino")
    End Sub

    Private Sub myMsg()
        MsgBox("Anda belum memilih jenis Arduino!
Silahkan pilih jenis Arduino terlebih dahulu.",
MsgBoxStyle.Critical, "Pilih Arduino")
    End Sub

    Private Sub selectArduino()
        'Get Selected Arduino
        If RB_Uno.Checked Then
            arduinoSelected = 1
        ElseIf RB_Mega.Checked Then
            arduinoSelected = 2
        ElseIf RB_Due.Checked Then
            arduinoSelected = 3
        End If
    End Sub

```

```

        'MsgBox(arduinoSelected.ToString)
    End Sub

    Private Sub Label2_Click(sender As Object, e
As EventArgs) Handles Label2.Click

        End Sub
    End Class

*Program pilih output
    Public Class Select_Output

        Private Sub Label2_Click(sender As Object, e
As EventArgs) Handles Label2.Click

            End Sub

            Private Sub Button2_Click(sender As Object, e
As EventArgs) Handles Button2.Click
                outputSelected = 0
                SelectOutput()
                If outputSelected = 0 Then
                    myMsg2()
                Else
                    Me.Visible = False
                End If

                'MsgBox(solenoid & " ," & lcd & " ," &
motor & " ," & segment & " ," & led,
MsgBoxStyle.Information, "")

                Dim sensor As String
                Dim ard As String
                Dim out As String
                sensor = ""
                ard = ""
                out = ""
                '===== Sensor Selected
=====

                If dht11 = 1 Then
                    sensor = sensor & " - DHT11" & vbCrLf
                End If
                If fingerprint = 1 Then

```

```

        sensor = sensor & " - Fingerprint" &
vbCrLf
    End If
    If lm35 = 1 Then
        sensor = sensor & " - LM35" & vbCrLf
    End If
    If level = 1 Then
        sensor = sensor & " - Level" & vbCrLf
    End If
    If ldr = 1 Then
        sensor = sensor & " - LDR" & vbCrLf
    End If

    '=====          Arduino          Selected
=====
    Select Case (arduinoSelected)
        Case 1
            ard = " - Arduino Uno" & vbCrLf
        Case 2
            ard = " - Arduino Mega" & vbCrLf
        Case 3
            ard = " - Arduino Due" & vbCrLf
    End Select

    '=====          Output          Selected
=====
    If selenoid = 1 Then
        out = out & " - Selenoid" & vbCrLf
    End If
    If lcd = 1 Then
        out = out & " - LCD 16x2" & vbCrLf
    End If
    If motor = 1 Then
        out = out & " - Motor DC" & vbCrLf
    End If
    If segment = 1 Then
        out = out & " - Segment" & vbCrLf
    End If
    If led = 1 Then
        out = out & " - LED" & vbCrLf
    End If

```

```

Form1.Configuration_window1.TXT_Configuration.Text =
"Konfigurasi yang telah Anda lakukan adalah sebagai
berikut :" & vbCrLf & _
    "Sensor : " & vbCrLf & _
    sensor & _
    "Arduino : " & vbCrLf & _
    ard & _
    "Output :" & vbCrLf & _
    out
End Sub

Private Sub Select_Output_Load(sender As
Object, e As EventArgs) Handles MyBase.Load
    outputSelected = 0

    'Output
    selenoid = 0
    lcd = 0
    motor = 0
    segment = 0
    led = 0

End Sub

Private Sub myMsg()
    MsgBox("Anda belum memilih jenis Output!
Silahkan pilih jenis Output terlebih dahulu",
MsgBoxStyle.Critical, "Pilih Output")
End Sub

Private Sub myMsg2()
    MsgBox("Untuk saat ini, Anda harus
memilih Output supaya bisa ke langkah berikutnya!",
MsgBoxStyle.Critical, "Pilih Output")
End Sub

Private Sub SelectOutput()
    'Get Selected Sensor

    If RB_Selenoid.Checked Then
        outputSelected = 1
        selenoid = 1
    Else

```



```

        selenoid = 0
    End If

    If RB_LCD.Checked Then
        outputSelected = 2
        lcd = 1
    Else
        lcd = 0
    End If

    If RB_Motor_DC.Checked Then
        outputSelected = 3
        motor = 1
    Else
        motor = 0
    End If

    If RB_7Segment.Checked Then
        outputSelected = 4
        segment = 1
    Else
        segment = 0
    End If

    If RB_LED.Checked Then
        outputSelected = 5
        led = 1
    Else
        led = 0
    End If

    If Not (RB_Solenoid.Checked Or
    RB_LCD.Checked Or RB_Motor_DC.Checked Or
    RB_7Segment.Checked Or RB_LED.Checked) Then
        outputSelected = 0
    End If

End Sub

Private Sub Button1_Click(sender As Object, e
As EventArgs) Handles Button1.Click
    Form1.SelectArduino1.Visible = True

End Sub

```

```

        Private Sub PB_ViewDBArduino_Click(sender As
Object, e As EventArgs) Handles PB_ViewDBArduino.Click
            outputSelected = 0
            SelectOutput()
            menuSelected = 2
            If outputSelected = 0 Then
                myMsg()
            Else
                view_datasheet.Show()
            End If
        End Sub
    End Class

```

***Program jendela konfigurasi**

```

Public Class Configuration_window

    Private Sub TextBox1_TextChanged(sender As
Object, e As EventArgs) Handles
TXT_Configuration.TextChanged

        End Sub

    Private Sub Configuration_window_Load(sender
As Object, e As EventArgs) Handles MyBase.Load

        'MsgBox(dht11 & ", " & fingerprint & ", "
& lm35 & ", " & level & ", " & ldr,
MsgBoxStyle.Information, "")
        End Sub

    Private Sub Button1_Click(sender As Object, e
As EventArgs) Handles Button1.Click
        Form1.Select_Output1.Visible = True
    End Sub

    Private Sub Button2_Click(sender As Object, e
As EventArgs) Handles Button2.Click

        Me.Visible = False
        Dim lib_ As String
        lib_ = ""
        If dht11 = 1 Then

```

```

        lib_ = lib_ & "- Library untuk Sensor
DHT11 " & vbCrLf & vbCrLf
    End If
    If fingerprint = 1 Then
        lib_ = lib_ & "- Library untuk Sensor
Fingerprint" & vbCrLf & vbCrLf
    End If
    If lm35 = 1 Then
        lib_ = lib_ & "- Library untuk Sensor
LM35" & vbCrLf & vbCrLf
    End If
    If lcd = 1 Then
        lib_ = lib_ & "- Library untuk Sensor
LCD 16x2" & vbCrLf & vbCrLf
    End If
    If lib_ = "" Then

Form1.Install_Library_Window1.Label2.Text = "Tidak ada
Library yang perlu diinstall."

Form1.Install_Library_Window1.Button3.Enabled = False
    Else

Form1.Install_Library_Window1.Button3.Enabled = True

Form1.Install_Library_Window1.Label2.Text = lib_
    End If

    End Sub
End Class

```

***Program jendela pasang library**

```

<Global.Microsoft.VisualBasic.CompilerServices.De
signerGenerated()> _
    Partial Class Install_Library_Window
        Inherits System.Windows.Forms.UserControl

        'UserControl overrides dispose to clean up
the component list.
        <System.Diagnostics.DebuggerNonUserCode()> _
            Protected Overrides Sub Dispose(ByVal
disposing As Boolean)
                Try

```

```

        If disposing AndAlso components IsNot
Nothing Then
            components.Dispose()
        End If
    Finally
        MyBase.Dispose(disposing)
    End Try
End Sub

'Required by the Windows Form Designer
Private components As
System.ComponentModel.IContainer

'NOTE: The following procedure is required by
the Windows Form Designer
'It can be modified using the Windows Form
Designer.
'Do not modify it using the code editor.
<System.Diagnostics.DebuggerStepThrough()> _
Private Sub InitializeComponent()
    Me.Button1 = New
System.Windows.Forms.Button()
    Me.Button2 = New
System.Windows.Forms.Button()
    Me.GroupBox1 = New
System.Windows.Forms.GroupBox()
    Me.Button3 = New
System.Windows.Forms.Button()
    Me.Label4 = New
System.Windows.Forms.Label()
    Me.Label2 = New
System.Windows.Forms.Label()
    Me.Label1 = New
System.Windows.Forms.Label()
    Me.GroupBox1.SuspendLayout()
    Me.SuspendLayout()
    '
    'Button1
    '
    Me.Button1.Font = New
System.Drawing.Font("Microsoft Sans Serif", 9.75!,
System.Drawing.FontStyle.Regular,
System.Drawing.GraphicsUnit.Point, CType(0, Byte))

```

```

        Me.Button1.Location = New
System.Drawing.Point(306, 222)
        Me.Button1.Name = "Button1"
        Me.Button1.Size = New
System.Drawing.Size(75, 25)
        Me.Button1.TabIndex = 30
        Me.Button1.Text = "Prev"
        Me.Button1.UseVisualStyleBackColor = True
    ,
    'Button2
    ,
        Me.Button2.Font = New
System.Drawing.Font("Microsoft Sans Serif", 9.75!,
System.Drawing.FontStyle.Regular,
System.Drawing.GraphicsUnit.Point, CType(0, Byte))
        Me.Button2.Location = New
System.Drawing.Point(387, 222)
        Me.Button2.Name = "Button2"
        Me.Button2.Size = New
System.Drawing.Size(75, 25)
        Me.Button2.TabIndex = 29
        Me.Button2.Text = "Next"
        Me.Button2.UseVisualStyleBackColor = True
    ,
    'GroupBox1
    ,
        Me.GroupBox1.Controls.Add(Me.Button3)
        Me.GroupBox1.Controls.Add(Me.Label4)
        Me.GroupBox1.Controls.Add(Me.Label2)
        Me.GroupBox1.Location = New
System.Drawing.Point(39, 38)
        Me.GroupBox1.Name = "GroupBox1"
        Me.GroupBox1.Size = New
System.Drawing.Size(423, 164)
        Me.GroupBox1.TabIndex = 28
        Me.GroupBox1.TabStop = False
        Me.GroupBox1.Text = "Library yang
dibutuhkan : "
    ,
    'Button3
    ,
        Me.Button3.Font = New
System.Drawing.Font("Microsoft Sans Serif", 9.75!,

```

```

System.Drawing.FontStyle.Regular,
System.Drawing.GraphicsUnit.Point, CType(0, Byte))
    Me.Button3.Location = New
System.Drawing.Point(287, 133)
    Me.Button3.Name = "Button3"
    Me.Button3.Size = New
System.Drawing.Size(130, 25)
    Me.Button3.TabIndex = 31
    Me.Button3.Text = "Cara Install"
    Me.Button3.UseVisualStyleBackColor = True
    '
    'Label4
    '
    Me.Label4.AutoSize = True
    Me.Label4.Location = New
System.Drawing.Point(16, 26)
    Me.Label4.Name = "Label4"
    Me.Label4.Size = New
System.Drawing.Size(230, 13)
    Me.Label4.TabIndex = 2
    Me.Label4.Text = "Library yang diutuhkan
adalah sebagai berikut : "
    '
    'Label2
    '
    Me.Label2.AutoSize = True
    Me.Label2.Location = New
System.Drawing.Point(29, 48)
    Me.Label2.Name = "Label2"
    Me.Label2.Size = New
System.Drawing.Size(142, 13)
    Me.Label2.TabIndex = 0
    Me.Label2.Text = "Library untuk Sensor
DHT11"
    '
    'Label1
    '
    Me.Label1.AutoSize = True
    Me.Label1.Font = New
System.Drawing.Font("Microsoft Sans Serif", 12.0!,
System.Drawing.FontStyle.Bold,
System.Drawing.GraphicsUnit.Point, CType(0, Byte))
    Me.Label1.Location = New
System.Drawing.Point(189, 15)

```

```

        Me.Label1.Name = "Label1"
        Me.Label1.Size = New
System.Drawing.Size(128, 20)
        Me.Label1.TabIndex = 27
        Me.Label1.Text = "Pasang Library"
    ,
    'Install_Library_Window
    ,
        Me.AutoScaleDimensions = New
System.Drawing.SizeF(6.0!, 13.0!)
        Me.AutoScaleMode =
System.Windows.Forms.AutoScaleModeMode.Font
        Me.Controls.Add(Me.Button1)
        Me.Controls.Add(Me.Button2)
        Me.Controls.Add(Me.GroupBox1)
        Me.Controls.Add(Me.Label1)
        Me.Name = "Install_Library_Window"
        Me.Size = New System.Drawing.Size(500,
270)
        Me.GroupBox1.ResumeLayout(False)
        Me.GroupBox1.PerformLayout()
        Me.ResumeLayout(False)
        Me.PerformLayout()

    End Sub
    Friend WithEvents Button1 As
System.Windows.Forms.Button
    Friend WithEvents Button2 As
System.Windows.Forms.Button
    Friend WithEvents GroupBox1 As
System.Windows.Forms.GroupBox
    Friend WithEvents Label1 As
System.Windows.Forms.Label
    Friend WithEvents Button3 As
System.Windows.Forms.Button
    Friend WithEvents Label4 As
System.Windows.Forms.Label
    Friend WithEvents Label2 As
System.Windows.Forms.Label

```

End Class

***Program jendela pasang program**

```
Public Class Install_Program_Window
```

```

        Private Sub Button3_Click(sender As Object, e
As EventArgs) Handles Button3.Click
            Install_Program.TextBox1.Text = ""
            Dim header As String
            Dim setup As String
            Dim loop_ As String
            Dim end_ As String

            header = Install_Program.txt_header.Text
            setup = Install_Program.txt_setup.Text
            loop_ = Install_Program.txt_loop.Text
            end_ = Install_Program.txt_end.Text
            If lm35 = 1 Then
                header = header & vbCrLf & _
                    "#include <lm35.h>" & vbCrLf & _
                    "lm35 temps(0); //A0" & vbCrLf & _
                loop_ = loop_ & vbCrLf & _
                    "===== LM35 =====" & vbCrLf & _
                    "temps.MeasureTemp(); // start
measuring temperature" & vbCrLf & _
                    "Serial.print(temps.TempInCelcius);
// print temperature" & vbCrLf & _
                    "Serial.println(\"C\");" & vbCrLf & _
                _
                "Serial.print(temps.TempInFahrenheit); // print temp
in Fahrenheit" & vbCrLf & _
                    "Serial.println(\"F\");" & vbCrLf & _
                _
                "delay(500); // delay of 5 seconds
between readings" & vbCrLf

                'Install_Program.TextBox1.Text =
Install_Program.TextBox1.Text & vbCrLf & _
                '                                     header &
Install_Program.txt_setup.Text & vbCrLf & loop_ & _
                '                                     Install_Program.txt_end.Text
            End If
            If level = 1 Then
                loop_ = loop_ & vbCrLf & _
                    "//=====                                LEVEL
===== " & vbCrLf & _

```



```

                                "   int level = analogRead(A1);" &
vbCrLf & _
                                "   Serial.print("Level  :");" &
vbCrLf & _
                                "       Serial.println(level);" &
vbCrLf

                                'Install_Program.TextBox1.Text      =
Install_Program.TextBox1.Text & _
                                '       Install_Program.txt_header.Text
& vbCrLf & Install_Program.txt_setup.Text & vbCrLf &
vbCrLf & _
                                '                                   loop_      &
Install_Program.txt_end.Text
                                End If
                                'Here --->
                                If ldr = 1 Then
                                    loop_ = loop_ & vbCrLf & _
                                        "//===== LDR ====="
& vbCrLf & _
                                    "   int ldr = analogRead(A2);" &
vbCrLf & _
                                    "       Serial.print("Intensitas
cahaya :");" & vbCrLf & _
                                    "       Serial.println(ldr);" & vbCrLf

                                'Install_Program.TextBox1.Text      =
Install_Program.TextBox1.Text & _
                                '       Install_Program.txt_header.Text
& vbCrLf & Install_Program.txt_setup.Text & vbCrLf &
vbCrLf & _
                                '                                   loop_      &
Install_Program.txt_end.Text
                                End If
                                If motor = 1 Then
                                    loop_ = loop_ & vbCrLf & _
                                        "//===== MOTOR =====" &
vbCrLf & _
                                    "   digitalWrite(13,LOW); //Motor
OFF" & vbCrLf & _
                                    "   delay(3000);" & vbCrLf &
                                    "   digitalWrite(13,HIGH); //Motor
ON" & vbCrLf

```

```

        'Install_Program.TextBox1.Text      =
Install_Program.TextBox1.Text & _
        '      Install_Program.txt_header.Text
& vbCrLf & Install_Program.txt_setup.Text & vbCrLf &
vbCrLf & _
        '                                  loop_      &
Install_Program.txt_end.Text
    End If
    If segment = 1 Then
        header = header & vbCrLf & _
            "//== 7 Segment and 74ls47 ===" &
vbCrLf & _
            "int  inputs[4]  = {4,5,6,7}; //"
A,B,C,D inputs" & vbCrLf & _
            "byte  BCD[16][4]  ={{0,0,0,0}," &
vbCrLf & _
            "{1,0,0,0}," & vbCrLf & _
            "{0,1,0,0}," & vbCrLf & _
            "{1,1,0,0}," & vbCrLf & _
            "{0,0,1,0}," & vbCrLf & _
            "{1,0,1,0}," & vbCrLf & _
            "{0,1,1,0}," & vbCrLf & _
            "{1,1,1,0}," & vbCrLf & _
            "{0,0,0,1}," & vbCrLf & _
            "{1,0,0,1}," & vbCrLf & _
            "{0,1,0,1}," & vbCrLf & _
            "{1,1,0,1}," & vbCrLf & _
            "{0,0,1,1}," & vbCrLf & _
            "{1,0,1,1}," & vbCrLf & _
            "{0,1,1,1}," & vbCrLf & _
            "{1,1,1,1}}; //BCD code" & vbCrLf
& _
            "int  number = 5; //which  number
in BCD code do you want to send" & vbCrLf

        setup = setup & vbCrLf & _
            "//Init pin 7 segment" & vbCrLf & _
            "for(int a = 0; a < 4; a++){" &
vbCrLf & _
            "    pinMode(inputs[a], OUTPUT);"
& vbCrLf & _
            "}" //set outputs" & vbCrLf

        loop_ = loop_ & vbCrLf & _

```

```

        "//Untuk 7 Segment Display dengan IC
74LS47" & vbCrLf & _
        "for(int c = 0; c < 4; c++){ " &
vbCrLf & _
        "                digitalWrite(inputs[c],
BCD[number][c]);" & vbCrLf & _
        "        delay(1000);" & vbCrLf & _
        "}" & vbCrLf

        'Install_Program.TextBox1.Text          =
Install_Program.TextBox1.Text & vbCrLf & _
        '        header & vbCrLf & _
        '        setup & vbCrLf & _
        '                                loop_      &
Install_Program.txt_end.Text
        End If
        If led = 1 Then
            loop_ = loop_ & vbCrLf & _
                "//===== LED ====="
& vbCrLf & _
                "        analogWrite(A3, 255);" &
vbCrLf & _
                "        delay(3000);" & vbCrLf & _
                "        analogWrite(A3, 1023);" &
vbCrLf & _
                "        delay(3000);" & vbCrLf
        End If
        Install_Program.TextBox1.Text          =
Install_Program.TextBox1.Text & _
            header & vbCrLf & setup & vbCrLf
& _
            loop_ & end_
        Install_Program.Show()

    End Sub

    Private Sub Button1_Click(sender As Object, e
As EventArgs) Handles Button1.Click
        Form1.Install_Library_Window1.Visible    =
True

    End Sub

```

```

Private Sub Button2_Click(sender As Object, e
As EventArgs) Handles Button2.Click
    Me.Visible = False
    Dim txt1 As String
    Dim txt2 As String
    Dim in_ As String
    Dim out_ As String
    Dim i As Integer
    Dim o As Integer
    in_ = ""
    out_ = ""
    i = 0
    o = 0
    txt1 = "Berikut ini merupakan koneksi
dari sensor ke pin Arduino ke output." & vbCrLf & _
        "Hubungkan koneksi dari sensor ke
pin Arduino sesuai dengan koneksi berikut ini :" &
vbCrLf
    txt2 = "Hubungkan koneksi dari pin
Arduino ke output sesuai dengan koneksi berikut ini :"
& vbCrLf

    If dht11 = 1 Then
        i = i + 1
        in_ = in_ & i & ". Data Sensor DHT
masuk ke Arduino D2." & vbCrLf
    End If
    If fingerprint = 1 Then
        i = i + 1
        in_ = in_ & i & ". Sensor Fingerprint
masuk ke Arduino D10 dan D11." & vbCrLf
    End If
    If lm35 = 1 Then
        i = i + 1
        in_ = in_ & i & ". Hubungkan Data
LM35 ke pin A0 Arduino" & vbCrLf
    End If
    If level = 1 Then
        i = i + 1
        in_ = in_ & i & ". Hubungkan output
dari Sensor Level ke pin A1 Arduino" & vbCrLf
    End If
    If ldr = 1 Then
        i = i + 1

```

```

        in_ = in_ & i & ". Hubungkan output
dari Sensor LDR ke pin A2 Arduino" & vbCrLf
    End If

    If selenoid = 1 Then
        o = o + 1
        out_ = out_ & o & ". Selenoid masuk
ke pin Arduino D9. " & vbCrLf
    End If
    If lcd = 1 Then
        o = o + 1
        out_ = out_ & o & ". Pin I2C LCD ke
Arduino (pin A4/SDA dan A5/SCL). " & vbCrLf
    End If
    If motor = 1 Then
        o = o + 1
        out_ = out_ & o & ". Hubungkan pin
Motor ke pin D13 Arduino" & vbCrLf
    End If
    If segment = 1 Then
        o = o + 1
        out_ = out_ & o & ". Hubungkan pin IC
Decoder 74LS47 (A,B,C,D) ke pin (D4,D5,D6,D7) Arduino"
& vbCrLf
    End If
    If led = 1 Then
        o = o + 1
        out_ = out_ & o & ". Hubungkan pin
LED ke pin A3 Arduino" & vbCrLf
    End If

    txt1 = txt1 & in_
    txt2 = txt2 & out_

    Form1.SettingPin1.TextBox1.Text = txt1 &
vbCrLf & txt2

End Sub

Private Sub Button4_Click(sender As Object, e
As EventArgs) Handles Button4.Click
    Install_Enroll.Show()
End Sub
End Class

```

***Program jendela program enroll**

```
<Global.Microsoft.VisualBasic.CompilerServices.DesignerGenerated()> _
Partial Class Install_Enroll
    Inherits System.Windows.Forms.Form

    'Form overrides dispose to clean up the component list.
    <System.Diagnostics.DebuggerNonUserCode()> _
    Protected Overrides Sub Dispose(ByVal disposing As Boolean)
        Try
            If disposing AndAlso components IsNot Nothing Then
                components.Dispose()
            End If
        Finally
            MyBase.Dispose(disposing)
        End Try
    End Sub

    'Required by the Windows Form Designer
    Private components As System.ComponentModel.IContainer

    'NOTE: The following procedure is required by the Windows Form Designer
    'It can be modified using the Windows Form Designer.
    'Do not modify it using the code editor.
    <System.Diagnostics.DebuggerStepThrough()> _
    Private Sub InitializeComponent()
        Dim resources As System.ComponentModel.ComponentResourceManager = New System.ComponentModel.ComponentResourceManager(GetType(Install_Enroll))
        Me.Button1 = New System.Windows.Forms.Button()
        Me.Button2 = New System.Windows.Forms.Button()
        Me.Label1 = New System.Windows.Forms.Label()
        Me.TextBox1 = New System.Windows.Forms.TextBox()
```

```

        Me.SuspendLayout()
        '
        'Button1
        '
        Me.Button1.Font = New
System.Drawing.Font("Microsoft Sans Serif", 9.75!,
System.Drawing.FontStyle.Regular,
System.Drawing.GraphicsUnit.Point, CType(0, Byte))
        Me.Button1.Location = New
System.Drawing.Point(336, 265)
        Me.Button1.Name = "Button1"
        Me.Button1.Size = New
System.Drawing.Size(75, 25)
        Me.Button1.TabIndex = 30
        Me.Button1.Text = "Close"
        Me.Button1.UseVisualStyleBackColor = True
        '
        'Button2
        '
        Me.Button2.Font = New
System.Drawing.Font("Microsoft Sans Serif", 9.75!,
System.Drawing.FontStyle.Regular,
System.Drawing.GraphicsUnit.Point, CType(0, Byte))
        Me.Button2.Location = New
System.Drawing.Point(417, 265)
        Me.Button2.Name = "Button2"
        Me.Button2.Size = New
System.Drawing.Size(75, 25)
        Me.Button2.TabIndex = 29
        Me.Button2.Text = "Copy"
        Me.Button2.UseVisualStyleBackColor = True
        '
        'Label1
        '
        Me.Label1.AutoSize = True
        Me.Label1.Font = New
System.Drawing.Font("Microsoft Sans Serif", 9.75!,
System.Drawing.FontStyle.Regular,
System.Drawing.GraphicsUnit.Point, CType(0, Byte))
        Me.Label1.Location = New
System.Drawing.Point(185, 20)
        Me.Label1.Name = "Label1"
        Me.Label1.Size = New
System.Drawing.Size(134, 16)

```

```

        Me.Label1.TabIndex = 28
        Me.Label1.Text = "Install Program Enroll"
    ,
    'TextBox1
    ,
        Me.TextBox1.Location = New
System.Drawing.Point(12, 51)
        Me.TextBox1.Multiline = True
        Me.TextBox1.Name = "TextBox1"
        Me.TextBox1.ScrollBars =
System.Windows.Forms.ScrollBars.Vertical =
        Me.TextBox1.Size = New
System.Drawing.Size(480, 207)
        Me.TextBox1.TabIndex = 27
        Me.TextBox1.Text =
resources.GetString("TextBox1.Text")
    ,
    'Install_Enroll
    ,
        Me.AutoScaleDimensions = New
System.Drawing.SizeF(6.0!, 13.0!)
        Me.AutoScaleMode =
System.Windows.Forms.AutoScaleMode.Font =
        Me.ClientSize = New
System.Drawing.Size(504, 302)
        Me.Controls.Add(Me.Button1)
        Me.Controls.Add(Me.Button2)
        Me.Controls.Add(Me.Label1)
        Me.Controls.Add(Me.TextBox1)
        Me.Name = "Install_Enroll"
        Me.StartPosition =
System.Windows.Forms.FormStartPosition.CenterScreen =
        Me.Text = "Install Program Enroll"
        Me.TopMost = True
        Me.ResumeLayout(False)
        Me.PerformLayout()

End Sub
Friend WithEvents Button1 As
System.Windows.Forms.Button
Friend WithEvents Button2 As
System.Windows.Forms.Button
Friend WithEvents Label1 As
System.Windows.Forms.Label

```



```

        Friend WithEvents TextBox1 As
System.Windows.Forms.TextBox
    End Class

```

***Program jendela seting pin**

```

Public Class SettingPin

```

```

    Private Sub Button2_Click(sender As Object, e
As EventArgs) Handles Button2.Click
        'checkPin()
        'If inputPin = "" Then
            '    MsgBox("You have not selected the
Pins yet, Please select for Pins!",
MsgBoxStyle.Critical, "Select Pin")
        'Else
            '    Me.Visible = False
        'End If
        Me.Visible = False
    End Sub

```

```

    Private Sub Button1_Click(sender As Object, e
As EventArgs) Handles Button1.Click
        Form1.Install_Program_Window1.Visible =
True
    End Sub

```

```

    Private Sub CB_D0_CheckedChanged(sender As
Object, e As EventArgs)

    End Sub

```

```

    Private Sub SettingPin_Load(sender As Object,
e As EventArgs) Handles MyBase.Load

    End Sub

```

```

    Private Sub checkPin()
        'If CB_D0.Checked Then
            '    inputPin = inputPin & "D0#"
        'End If
        'If CB_D1.Checked Then
            '    inputPin = inputPin & "D1#"
        'End If
        'If CB_D2.Checked Then

```

```

        '    inputPin = inputPin & "D2#"
    'End If
    'If CB_D3.Checked Then
        '    inputPin = inputPin & "D3#"
    'End If
    'If CB_D4.Checked Then
        '    inputPin = inputPin & "D4#"
    'End If
End Sub

Private Sub TextBox1_TextChanged(sender As
Object, e As EventArgs) Handles TextBox1.TextChanged
    Dim txt As String

    txt = "Rangkai Sensor DHT11, Sensor
Fingerprint, Relay, dan LCD 16x4 sebagai berikut :" &
vbCrLf

End Sub
End Class

```

***Program jendela terakhir**

```

<Global.Microsoft.VisualBasic.CompilerServices.De
signerGenerated()> _
Partial Class Finish
    Inherits System.Windows.Forms.UserControl

    'UserControl overrides dispose to clean up
the component list.
    <System.Diagnostics.DebuggerNonUserCode()> _
    Protected Overrides Sub Dispose(ByVal
disposing As Boolean)
        Try
            If disposing AndAlso components IsNot
Nothing Then
                components.Dispose()
            End If
        Finally
            MyBase.Dispose(disposing)
        End Try
    End Sub

    'Required by the Windows Form Designer

```

```

        Private components As
        System.ComponentModel.IContainer

        'NOTE: The following procedure is required by
        the Windows Form Designer
        'It can be modified using the Windows Form
        Designer.
        'Do not modify it using the code editor.
        <System.Diagnostics.DebuggerStepThrough()> _
        Private Sub InitializeComponent()
            Me.Label1 = New
            System.Windows.Forms.Label()
            Me.Button1 = New
            System.Windows.Forms.Button()
            Me.Button2 = New
            System.Windows.Forms.Button()
            Me.GroupBox1 = New
            System.Windows.Forms.GroupBox()
            Me.Label2 = New
            System.Windows.Forms.Label()
            Me.GroupBox1.SuspendLayout()
            Me.SuspendLayout()
            '
            'Label1
            '
            Me.Label1.AutoSize = True
            Me.Label1.Font = New
            System.Drawing.Font("Microsoft Sans Serif", 12.0!,
            System.Drawing.FontStyle.Bold,
            System.Drawing.GraphicsUnit.Point, CType(0, Byte))
            Me.Label1.Location = New
            System.Drawing.Point(216, 15)
            Me.Label1.Name = "Label1"
            Me.Label1.Size = New
            System.Drawing.Size(68, 20)
            Me.Label1.TabIndex = 13
            Me.Label1.Text = "Selesai"
            '
            'Button1
            '
            Me.Button1.Font = New
            System.Drawing.Font("Microsoft Sans Serif", 9.75!,
            System.Drawing.FontStyle.Regular,
            System.Drawing.GraphicsUnit.Point, CType(0, Byte))

```

```

        Me.Button1.Location = New
System.Drawing.Point(296, 222)
        Me.Button1.Name = "Button1"
        Me.Button1.Size = New
System.Drawing.Size(75, 25)
        Me.Button1.TabIndex = 26
        Me.Button1.Text = "Prev"
        Me.Button1.UseVisualStyleBackColor = True
        ,
        'Button2
        ,
        Me.Button2.Font = New
System.Drawing.Font("Microsoft Sans Serif", 9.75!,
System.Drawing.FontStyle.Regular,
System.Drawing.GraphicsUnit.Point, CType(0, Byte))
        Me.Button2.Location = New
System.Drawing.Point(377, 222)
        Me.Button2.Name = "Button2"
        Me.Button2.Size = New
System.Drawing.Size(85, 25)
        Me.Button2.TabIndex = 25
        Me.Button2.Text = "Close"
        Me.Button2.UseVisualStyleBackColor = True
        ,
        'GroupBox1
        ,
        Me.GroupBox1.Controls.Add(Me.Label2)
        Me.GroupBox1.Location = New
System.Drawing.Point(39, 38)
        Me.GroupBox1.Name = "GroupBox1"
        Me.GroupBox1.Size = New
System.Drawing.Size(423, 164)
        Me.GroupBox1.TabIndex = 27
        Me.GroupBox1.TabStop = False
        Me.GroupBox1.Text = "Catatan"
        ,
        'Label2
        ,
        Me.Label2.AutoSize = True
        Me.Label2.Location = New
System.Drawing.Point(20, 63)
        Me.Label2.Name = "Label2"
        Me.Label2.Size = New
System.Drawing.Size(382, 39)

```

```

        Me.Label2.TabIndex = 0
        Me.Label2.Text = "SELAMAT" &
Global.Microsoft.VisualBasic.ChrW(13) &
Global.Microsoft.VisualBasic.ChrW(10) & "Konfigurasi
yang telah anda lakukan sudah selesai. Sekarang
tinggal menc" & _
        "oba" &
Global.Microsoft.VisualBasic.ChrW(13) &
Global.Microsoft.VisualBasic.ChrW(10) & "di hardware-
nya." & Global.Microsoft.VisualBasic.ChrW(13) &
Global.Microsoft.VisualBasic.ChrW(10)
        ,
        'Finish
        ,
        Me.AutoScaleDimensions = New
System.Drawing.SizeF(6.0!, 13.0!)
        Me.AutoScaleMode =
System.Windows.Forms.AutoScaleMode.Font
        Me.Controls.Add(Me.GroupBox1)
        Me.Controls.Add(Me.Button1)
        Me.Controls.Add(Me.Label1)
        Me.Controls.Add(Me.Button2)
        Me.Name = "Finish"
        Me.Size = New System.Drawing.Size(500,
270)
        Me.GroupBox1.ResumeLayout(False)
        Me.GroupBox1.PerformLayout()
        Me.ResumeLayout(False)
        Me.PerformLayout()

End Sub
Friend WithEvents Label1 As
System.Windows.Forms.Label
Friend WithEvents Button1 As
System.Windows.Forms.Button
Friend WithEvents Button2 As
System.Windows.Forms.Button
Friend WithEvents GroupBox1 As
System.Windows.Forms.GroupBox
Friend WithEvents Label2 As
System.Windows.Forms.Label

```

```
End Class
```

-----Halaman ini sengaja dikosongkan-----

DAFTAR RIWAYAT HIDUP



Nama : Setyo Budi Utomo
NRP : 2214030052
Jenis Kelamin : Laki-laki
TTL : Surabaya, 21 April 1995
Agama : Islam
Alamat :
Jl. Tambak Dalam Baru 3/41
No. HP : 083831707601
E-mail :
setyo_utomo14@yahoo.com
Social Network : Abbud Yow Thyoo
Cita-cita : Pengusaha

RIWAYAT PENDIDIKAN

No	Instansi Pendidikan	Tahun
1	TK widya tama 2	2000-2001
2	SDN Asemrowo 1	2001-2007
3	SMPN 4 Surabaya	2007-2010
4	SMKN 5 Surabaya	2010-2014